School of Computer Science

COMP47360 - Research Practicum

Individual Report
Conor O'Kelly - 15203738

Student:	Conor O'Kelly
Student No:	15203738
Team:	T1 - Who's There
Team members:	Silvia Saloni, Ophélie Alliaume, Devin Stacey and Conor O'Kelly
GitHub link:	https://github.com/c-okelly/whothere
Website Link:	csi420-01-vm1.ucd.ie
Supervisor(s):	Prof. Pádraig Cunningham, Prof. Julie Berndsen, Dr. Anthony Ventresque, Dr. Gavin McArdle
Date:	August 26, 2016
Total Number of Pages:	64

Abstract

The objective of this report is to review the process and outcome of Team One's research practicum for the UCD's Computer Science conversion masters. This is the individual report submitted by Conor O'Kelly.

The group took the name "Who's There" as a play of words on the overall goal of the project. The team members comprised of Silvia Saloni, Ophélie Alliaume, Devin Stacey and Conor O'Kelly (Myself).

The main objective of the project was to estimate the number of people in a room and to ascertain the level of occupancy based on the devices connected to a Wifi access point. This would be achieved through the creation of a data model that maps the relationship between number of devices and individuals. The data was generated from UCD's campus wide wifi system and modelled against ground truth data collected manually. The results were required to be presented in the form of a web application. The website would be precise and easy to understand.

Each team was given the same problem statement and data. They were guided by customer feedback as they developed their software. Teams where required to make numerous technical decisions as to what features to implement and the process required to achieve them. This process needed teams to work together effectively to implement a complex software project.

Table Of Contents

1 Introduction

This reports gives an overview of the process employed by the development team and reviews the project's final outcome. The introduction will outline the problem statement and give an overview to each of the technical areas explored in the project outcome. Followed up by background research explored in the form of a literature review.

This details current research in the area of occupancy estimation in the building using automated tools. This is then proceeded by an in-depth look at my personal contribution to the project. Finally the report finishes by looking at a conclusion and future work.

The appendix of this report contains my profile, class blog and personal blogs. The extra documentation created by the team is included in the documentation directory in the code folder for this project. These documents were originally planned to be included in the appendix but this resulted in report being over 200 pages long as are placed elsewhere.

1.1 Problem Statement

UCD required an efficient cost effective system that would estimate the level of room utilisation across the campus. The current system costs approximately 25,000 Euros a year to run and requires individuals to manually check rooms on campus 15 minutes after classes have started. They estimate the levels of occupancy based on the following bins: 0% 25% 50% 75% and 100% This information is used by UCD to assist in the allocation of rooms.

Problems with the current system:

- High costs
- Short term observation
- Issues with estimating actual numbers of individuals in the room with only a short observation
- Labour intensive

The main objective of the project is to create an automated system that could generate this data efficiently, more cost effectively and less labour intensive for UCD's administration department.

To achieve the required outcome necessitated combing data into a model and then presenting our results. The raw data sources were as follows;

- Wifi logs of connected users specific to room level access points
- Timetables detailing modules and number of expected individuals.
- Two week period of surveys to act as ground trut

This information would first have to be cleaned and then correctly stored for later work.

The analysed data was then used to build our model. The data model would define the relationship between number of devices connected and the number of individuals present at the time. The results are then stored in the database.

A graphical interface was then used to present a clear overview of our results to users. This would take the form of a website application run by the server allocated to our team.

The team were also encouraged to make innovations based on the core requirements and feedback received from the customer. This is to add additional functionality to the overall product. These innovations separate out different teams.

1.2 Overview of overall outcome

This section will be a brief overview of the main areas that are looked at in the project outcome section.

The first section of main reports will look at the overall methods and success used in the management process of the project. The aim from the start was to use an agile methodology known as Scrum in order to guide the development. The group used Scrum as a template for our management process, using the features we found useful and productive. The report looks at the overall starting plan and how this worked out over the project.

This section is closely related to the coding guidelines and frameworks selected by the team. Strict formatting allowed collaborative coding to take place. The most crucial decisions were selecting formal coding standards and the processes of code review.

Next we examine the choice and development of the teams database. The team initially examined a number of options when selecting a database. After discussion and research it was decided to build a MySQL database using Oracles community server. This leads cleanly into the process of designing the schema for the database. Designs were based on the Software requirements specification as we aim to correctly map the data. This adapted over time in order to meet new requirements and as errors where discovered.

After this the report examines the selection and setup of the web server. Most teams decided to continue working with the familiar technology of Flask. Our team choose instead to develop new skills by building the server in Java using Spring. Part of the rationale was to achieve a scalable and professional server for our customers. The implementation success and problems are explored in this section.

The design of the database is clearly tied to the data it is required to store. The first required step is to clean the data before storage can take place. The choice of Python as the teams data processing language and design of the program layout is examined.

We then delve into the process of data analytics which brought about additional requirements. The team choose to use R as the language to implement the data analytics. The main models implemented are linear and logistic regressions. This formed a crucial corner stone of the overall project as the ability to predict accurate room occupancy levels was a

core requirement. The process of cleaning the data, developing the model and fine tuning is considered.

The next step was the display of information. Part of the teams innovation process was to use a design practice known as UX design. This framed our process of overall web design. The design of the website facilitated the display of the results from the overall process on our main page. The processes of turning raw data into an interactive map with the ability to specify different time periods and rooms is described. This section also included a small presentation on some of the functionality of the main page.

Once all the website and server was implemented it was important to carry out performance testing. A number of different methodologies were tested and analysed. The results and changes made are presented.

Penultimately we reviewed the numerous innovations by the team and their respective success. The most defining innovation was the successful deployment of an Android application to allow the gathering ground truth data. Additionally to this was the external API, UX testing processes and the ability to download dynamically created PDF reports for the customer. These reports give extra feedback about the data analysis based on the user's request.

Finally we show case the final product and its different functionality as we attempt to tie all the sections together. This concludes the main section of the report focusing on the outcome of the overall project.

1.3 My role within the team

The next major section examines my overall contribution to the project. I first examined the individual sections that I was responsible for and how these were carried out. I finished the section by looking at my overall role within the team as team coordinator. It was my responsibility to ensure that everyone was on task, meeting individual deadlines and progressing correctly. Effective project management was facilitated by a number of different tools as I outline.

1.4 Conclusion and future work

The reports finishes by conclusion of the overall project and looking a possible future work. A number of the innovations that the team had originally looked at became beyond the time scope of the original project. This section also looks at methods that might be used to improve the accuracy of our data analytics models.

2 Literature Review

Reviewing different methods to automatically assess the occupancy of rooms or buildings.

2.1 Abstract

Over the past decade the use of Wi-Fi has exploded. The dependency on Wi-Fi connections for both business and personal matters has lead to an 80% of smartphone users checking their phone within 15 minutes of waking up and 79% of smartphone users having their phone with them for 22 hours a day [IDC]. This dependency on wireless connections means that there is an incredible amount of data to be harvested from Wi-Fi connections.

The objective in this review is to assess methods used in estimating the occupancy of a room. The topics explored are the use of Wi-Fi logs, multi sensor networks, Wi-Fi signal degradation and OS fingerprinting.

This review allowed allow a technical context to be established for the current project.

2.2 Overview

The main purpose of this review is to outline the different methods that use automated techniques for estimating the number of people that are in a room. The research completed explores numerous methods of identifying occupancy in rooms or buildings. Currently the chosen methodology of Wifi logs details users connected at room specific access points. Alternative methods are examined to provide a comparison.

Gathering data on room occupancy provides important information to administrators. Rooms can be reallocated based on use, overall class attendance tracked and power management practice put in place.

The first topic of review explores similar methods to the use of Wifi logs as the basis for our data. This is followed by examining the use of multi sensor networks to provide additional or an independent data set. Subsequently techniques using Wi-Fi signal strength and degradation to estimated number of individuals in a space are examined.

The final topic of interest presents the success of identifying the use of OS fingerprinting to determine device type. Correctly estimating the ratio of phone to laptops identifies two important populations.

2.3 Wi-Fi logs

There are multiple methods available to automatically count the number of individuals in a room or area. The method chosen will depend on accuracy requirements and budget limitations.

Methods of measurement can be classified into three levels: [1]

- No additional hardware or software required. Using the current infrastructure and software that already exists.
- Addition of new software to existing hardware in order to generate the new data
- Addition of new software and hardware to the network in order to generate the data. (Generally in the form of a multi sensor network.)

The cheapest and easiest method will be to use existing equipment. A great example is Wi-Fi logs that track the number of devices connected at an access. Two main systems are used to track the number of devices connected to an access point [1]. The address resolution table (ARP) and the DHCP table. The ARP table is normally the more accurate of the two. In UCD, IP addresses are only given 30 minutes of lease time before they expire. Most likely logs in this project are derived from the ARP table.

This method is an effective way of counting the number of devices connected to an access point. However it gives no information as to the location of the user and or type of device.

This form of measurement has a number of issues. The first being the number of devices connected does not automatically translate into the number of individuals present. Depending on the population group the average number of devices will change. Often a bias will exist for certain populations to have higher averages [2]. A proposed example in UCD would be Computer Science students are more likely to have a phone and laptop as compared to students who are not required to have laptops. Another potential issue is laptops being prohibited during the use of some classes.

Another common problem that occurs is access points overlapping each other. All device types have their own system for switching between access points. This can result in individuals connected to access points in areas not geographically associated with them. A good example is a user connected to an access point on a different floor or outside the room. When looking for high accuracy in rooms this can cause problems as devices are attributed to the wrong location.

Overall this method of using Wi-Fi log data on its own can provide an effective rough estimate for the number of people in a given location. In the MIT example [2] we see they have easily identified overall trends in behavior throughout the campus.

It is the addition of extra information that can provide more accuracy to these estimates. These possible methods are examined in the next two sections.

2.4 Multi Sensor networks

The second type of methodology commonly examined falls under the third level of measurement. This is the addition of both hardware and software to the testing environment in order to get more accurate results. There are numerous different technologies that might be used. This would include but not be limited to the following; motion detectors, CO2 detectors, pressure sensors, lasers, acoustic sensors and cameras.

There has been significant research on using each of these type of technologies in order to track occupancy across buildings. In most of the research the main issue that is encountered is the high cost of implementing these systems. In order to effectively track people the whole building would need new hardware to be installed.

A study conducted using conditional random fields and a combination of sensors showed that these devices used together can be very effective measures at estimating the number of people gathered in a particular area [3]. In this study a combination of laser beams, pressure mat, CO2 detectors and PIR motion sensor were used. All of these devices combine in a non obtrusive way to gather information about the local occupancy levels. This was then used as a single estimation using a conditional random field model for real time information.

Overall this study showed that using a combination of sensors could predict with 90%+ accuracy levels while maintaining an acceptable level of the user's privacy. This method should be considered in most environments alongside estimation for the cost of implementing such a solution. One of the main reasons to pick a methodology like this would be where there is high cost associated with inaccurate predictions.

2.5 Wi-Fi signal degradation and change as a measure

Using changes in Wi-Fi signal can provide valuable information as to what is occurring locally. In the first paper [4] the processes are carried out using a sender and receiver. The pair are placed at opposite ends of the area to be measured. The receiver logs the signal strength and changes that occur over time. In this experiment people walk through the path and area between the pair of sensors.

This change in signal strength is a combination of direct block as an individual crosses the path and multi-path fading. In this experiment it was found that lower levels of occupancy have more pronounced effect on signal blocking while the multi-path is more affected by higher levels of occupancy [4]. It is the combination of both of these effects that gives accurate feedback.

The experiment looked at groups of 1-9 people. The estimation errors was 0 or 1 in 88% of the cases and 2 or less 100% of the time [4]. Overall this could be used as a good estimate for the number of people in an area. A more interesting impact of this study would be looking at larger groups of people. A number of studies in this area have shown that rough accuracy results can be generated to give general overall view [5].

2.6 OS Fingerprinting

The last section of this review looks at a method that could be used to characterise the two main categories of devices. The two categories of devices are smartphones and laptop. This information can be used to fine tune models for occupancy estimation and to give an insight into the type of activation that is carried out in an area.

The starting point for this kind of classification would be to look at the MAC address for devices and see if they could be identified that way. Most MAC address can easily be

attributed back to a single vendor unfortunately it is much more difficult to discover the type of device using this technique. While some vendors use specific methods for MAC address designation, most do not.

An alternative approach to OS fingerprinting can be found by looking at the DNS traffic that the device creates. This approach consisted of looking at a combination of the choice of domain name, query pattern and time intervals [6]. When looking at the data generated of a non-short period of time it was possible to get some estimation on the types of operating system being used.

2.7 Conclusion

The objective of this study has been to examine different methods of estimating occupancy. Three methods for numbers estimated and a fourth method for generating an additional data feature is examined. The methods were chosen due to their integral theme of using automated systems in order to generate information. Any system must be implementable with minimal need for human interaction to be scalable and automated.

The first method of using pre-existing Wi-Fi log data was the most straightforward and therefore most appealing. This method acts as a baseline which all other techniques can be compared to. In order to gain the most concise results it was noted that a large investment in multi-sensor networks is required. Additionally the gathering of extra information by looking for extra characteristics, such as Operating System, in the connected devices has been found to be valuable.

Depending on the required level of accuracy each of these methods showed promise in providing solid automated feedback about the occupancy of different locations. As the use of Wi-Fi logs has been the chosen method of approximation future work will include improving accuracy of this capacity estimation.

3 Project outcome

3.1 Management Process

Creating software is a complex endeavor requiring multiple elements to come together. According to a Garner survey carried out in 2012 nearly 50% of all IT projects end in some level of failure. It is crucial that a management process be implemented from the start to guide the team and provide structure. The process must be flexible enough to deal with technical setbacks while setting a realistic roadmap.

During the previous semester each member of the team used an Agile methodology known as Scrum to manage their projects. The Scrum system became the template by which the team self managed. This section outlines the basics of Scrum and the features our team choose to implement.

In Scrum, work is divided into a series of sprints, these form a roadmap to guide the team from the start to the finish of the development process. At each stage customer feedback is used to clarify and validate project progress, ensuring customer requirements' are met.

Each sprint has specific objectives that are broken down into a series of tasks or stories. This allows the team to maintain focus and commit to small, specific and tangible targets. Stories remain fixed throughout the sprint and the success of the team is reviewed at the sprint's conclusion. This encourages the team to reflect and improve throughout the process.

Crucial to the overall Scrum process is the use of shared tools for communication. Sprint backlogs and Trello allow tasks to be defined and committed to. Meeting minutes are used to clarify decisions and track overall progress. WhatsApp and Slack for direct communication and daily Scrum meetings allow the team to collaborate on issues and refocus objectives.

For coding, the team used a private GitHub repository for version control. This facilitated continuous integration of the codebase and tracking of issues and bugs.

During the first sprint the team created the following roadmap to guide development and set targets;

- Sprint 0 (week 2) => Development of SRS and exploring technologies
- Sprint 1 (week 3-4) => Decisions on technology choices and rough working prototype / Server / Database / basic website / cleaning scripts.
- Sprint 2 (week 5-6) => Data management / Data modelling / Model selection / server APIs / Website presentation
- Sprint 3 (week 7-8) => Extra website features / UX design / Android app development
- Sprint 4 (week 9) => Tying up loose end of the project
- Sprint 5 (week 10) => Report write up / final presentation / debugging / Extra issues

Each sprint had a number of objectives easily divided into a series of distinct tasks. Our roadmap helped the team focus on the finished product with a series of achievable steps. The team managed to finish the project on time using this as our guide. The main change was combining sprint 3 and 4 together. This occurred as the team underestimated the difficulty in implementing the data analytics modelling and finishing the website interface.

At conclusion the team met all core requirements and the majority of planned innovations. Backlogs played a crucial role by providing high level goals and targets throughout sprints. The most important process to our success was regular daily meetings and concise minutes of the project. These meetings provided a focal point through which the team could communicate and adapt as issues arose. As time went on the minutes partially took on the role of the sprint backlogs with task and overall goals being planned out in our meetings. (Included in the appendix are sections of minutes and sprint backlogs.)

A solid management process is second only to the creation of a cohesive team environment. This requires clear vision of overall goals and the ability to trust other members of the team. Team members must be able to acknowledge areas of weakness to allow for collaboration. It was this kind of environment that allowed our team to work as a solid unit throughout the research practicum.

3.2 Code guidelines / framework

Our system was built to possess the scope for using the best programming languages on the market for fulfilling its major tasks. This came with the cost of implementing many languages, which required for us to both know and implement several styles of programming syntax. In this context a coding style guideline was necessary for helping us to code in the most consistent way throughout all the project.

As a team we knew the importance of coding in the tidiest and most consistent way possible for ensuring that each member of our team could understand the code and work with it. A clean code is also easy to debug and to test.

Furthermore, this will ensure that our code could be potentially used by other developers to be extended and improved.

Apart from Python for which we used the PEP 8 Style Guide, for all the other languages used we followed Google's style guides. For viewing the complete style guide look at Appendix.

3.3 Database selection and design

Choosing the correct database technology and designing a robust but flexible schema were identified early on as high priority objectives for the team. Once committed to a database technology, the team invested significant time fine tuning our setup. This left us in a good position to adapt to new requirements easily as they arose.

Examining our main options three choices were identified; SQLite, MongoDB and MySQL. Our team carried out basic research on each database technology in order to make a final

decision. A flat file structure database was rejected due to efficiency concerns.

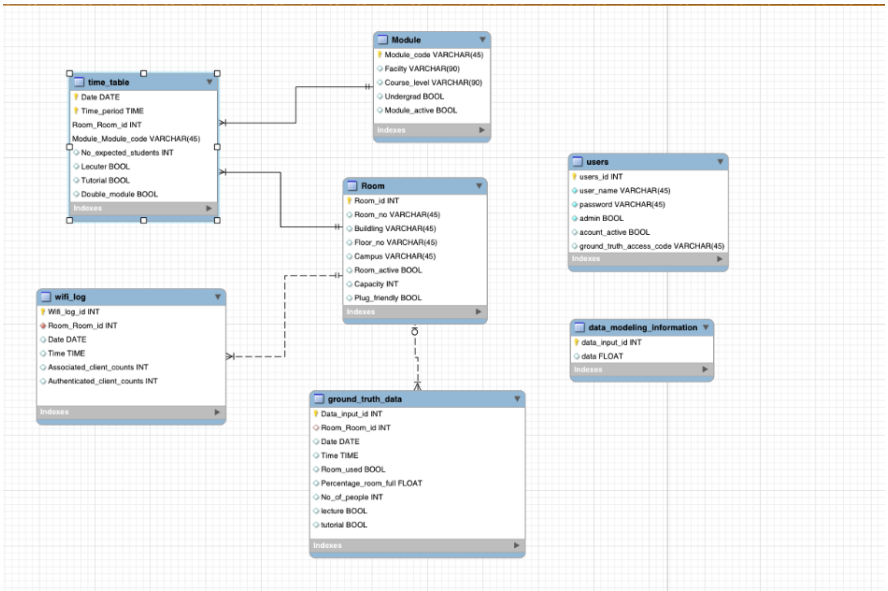
MongoDB was quickly ruled out as the database was not fully ACID compliant. While an unlikely issue at the project's proposed scale a larger commercial version of the software would need to meet full ACID standards.

SQLite, a technology familiar to much of the team, was ruled out due to concurrency write issues. With a large system with multiple information input sources (file data and Android app) this would become increasingly problematic.

The team decided to use a MySQL community server. This provided professional grade software that would allow the product to scale. The database would run as a dedicated process by providing connection through the localhost. This setup would also provide the team with valuable experience in commercial software use.

The team aimed to create a database schema to meet the requirements specified in our SRS document and by the data set. The design process of the schema would map the relationships between different data structures and achieve at least 2nd normal. The goal was to lose no information in the storage process.

The first draft of this is displayed below. In this schema the database ties all of the relevant information together and achieves second normal form. Relationships between tables are mapped using foreign keys. On the right hand side of the schema, tables are located that are not directly linked to the main data section but required by the software to function.

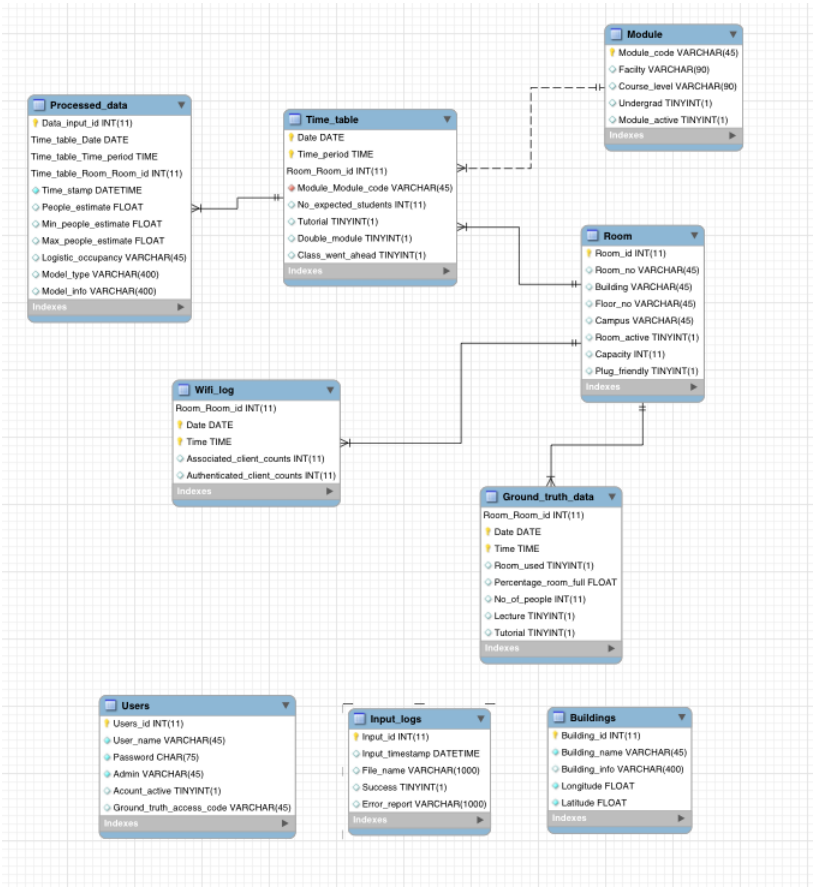


Over the development process the schema was changed to adapt to new requirements. Most changes were minor, the most significant alteration was the inclusion of data modelling output in the main structure.

During implementation a number of bugs were discovered, two stood out. First were different levels of strictness in case sensitivity for MySQL server versions. This threw a number of

confusing errors. A more serious bug issue was group-by statements being rejected on non linux versions but working fine on the server. The cause was attributed to different rules set being implemented for Group-by statements.

The final design of the database is displayed below. Overall the team found that working with Mysql to be quite easy and effective. In future a more open source version would be explored.



3.4 Spring Server - Choice and Set up

3.4.1 Choice of Spring

For our selection of Framework, we went through a number of debates on which would best suit our project, and which language it should be implemented with. When debating this, our team decided to utilise the Java language over Python, as a compiled language should outperform an Interpreted language in terms of speed, despite its drawbacks in increased complexity. Equally, as Java is considered an industry leader for server side development, it granted us a vast network of resources to research in order to implement our more complicated concepts. Equally, we followed up on this by selecting Spring as our choice of framework. While there were alternative choices, Spring offered us a variety of benefits which its competitors lacked.

Firstly, Spring had a very low entry point. As a lot of enterprises use Spring - including certain projects found within Amazon, Google, and eBay - there is plenty of documentation to aide in the development of systems which run off Spring. During the initial stages of selection, we found the guides on http://spring.io/guides to be very helpful in providing a baseline to begin our work. Equally, due to the sheer amount of guides Pivotal provides for Spring, future extension of this project should not be difficult for anybody who wishes to expand upon our final product.

Secondly, Spring is a Framework compatible with a variety of technologies. As it runs natively within Java, it can run any Java based libraries, alongside certain exclusives to the Spring platform. An example of the possibilities within Spring is that it is 100% compatible with Android through its Spring for Android dependencies, something we found that other Frameworks lacked. This means Android could be implemented into any existing server based upon the Spring Framework easily. As such, these features makes Spring a valuable resource when developing an application which would require a wide variety of technologies.

Finally, Spring is a Framework which offers modularity. At its core, Spring is a lightweight Framework which allows developers to select the components they wish to include in their project when required. While it is highly compatible, one is never forced to include all modules for all use cases. Instead, Spring has a pom.xml file which you may modify by including a GroupID, ArtefactID, and version number of any dependency you wish to include. This allows a developer to build a web application as small or as large as they wish.

3.4.2 Set Up

Initially, when setting up the Spring Framework, we built it as a Maven Project, which automatically structured it with specific directories. Images, CSS and Javascript could be placed within the Static Folder, HTML files could be included in the Templates Folder and your Java classes would be placed within a custom Java package, in our case, ie/ucd/serverjavafiles. The initial set up was focussed around two classes - Application.java and GreetingController.java. The Application.java class would initialise an instance of Apache Tomcat and allow access to the files we included in our Spring server. Secondly, the GreetingController.java class holds the configurations our group has mapped to resolve specific HTML files through a URL entered, alongside any additional code that should be executed when the method associated with that mapping is called. Following on this, we decided to implement the ApiController.java class which would handle any requests to URLs that would be associated with a RESTful web service alongside FileUploadController.java which had additional configurations relating to the uploading of new files.

As well as mapping URLs to certain HTML files or String output, we also had to configure our version of Spring to suit our needs if we wished to accomplish more complicated tasks. There are two kinds of configurations that we implemented during the set up stage of our Framework, Mvc Configurations as well as the Security Configuration. Mvc configurations are handled within the class MvcConfig.java which sets up a variety of helpful tools that our server utilises, such as the establishment of a data connection to our database by configuring a dataSource, creating a bean for SpringSecurityDialect to introduce .JSP functionality into HTML pages via Thymeleaf, alongside a general resolver for the login page. The security configurations we set up within WebSecurityConfig.java handle authentication within our environment, with the method “configAuthentication” establishing certain permissions for

our users through our login system and the method "configure" locking out certain webpages for unauthorised users and handling redirections for users not yet logged in.

Finally, there is an additional set of classes called upon by the methods mapped to individual URLs. There are a variety of classes which take in data from a POST form and store it to be utilised during the POST stage of a RequestMapping. Examples of this kind of class include Email.java, GroundTruthData.java, Registration.java as well as Upgrade.java. To show this kind of class working, an example would be the POST section of the /contact mapping which would create an instance of the SendMail.java class. This class contains the relevant code associated with JavaMailSenderImpl which would be able to send outgoing emails through a configuration file titled MailConfig.xml.

When the POST form is submitted, information such as message, sender, and name are stored within an instance of Email.java through the method "model.addAttribute()" which allows the data to be processed and sent by the SendMail.java class at a later stage. Other helpful classes are SqlQueries.java which contains all the relevant methods required to execute an SQL query from Java, which allows our project to perform relevant tasks such as checking if a user exists in our database, changing a user's status, adding a new user to the database, adding additional ground truth data via the Android application and fetching the relevant information required to build JSON Strings for our API.

3.5 Data phrasing and storage code

For data processing the team decided to use Python. The main motivation was familiarity with the language. Secondly was the large number of libraries for data phrasing and flexibility and ease of use of the language. A main concern with using Python was loss in efficiency from calling scripts using our server language Java. The team identified this concern but decided the benefits would outweigh the costs.

The data input manager was required to fulfil four main requirements.

- Generate the database from file if it does not already exist.
- Phrase and input data from three file types into the database.
- Generate feedback from the data input process.
- Generate derived data from existing information with the database.

The code was designed as a series of independent modules. A single module for phrasing each file type and a module responsible for generating derived data, creating the database and any additional features. These are managed by a controller module responsible for creating the database and populating it.

This ensures the database is first created before files are accessed and then processed using the correct module. This return is then populated into the database before derived information is then generated. An example of this is the facility school being derived from the module code. Finally the controller module provides feedback for each file storing this information in the database.

Files that are successfully processed were moved to a success directory with files that failed to be processed moved to a separate directory. This allows administrators to review problematic files easily.

This whole process can be launched manually or automatically when files are uploaded to the server.

3.6 Data analytics process

The aim of this analysis was to find a model that could accurately predict the number of people present in a room at any given hour by using the number of Wi-Fi logs registered in the same room and hour. In order to fulfil the aim of the analysis we used the dataset obtained by querying from the database only the information related to the ground truth data collected in room B.002, B.004 and B.006 from 9 to 17, the correspondent time table details and Wi-Fi associated logs. Wi-fi logs were summarised either as the average of logs counted within each hour for each room or as the maximum number of logs registered in each hour for each room. The dataset, so created had in total 216 rows and 12 columns.

Given the limited number of observations we decided only to run standard statistical analyses: linear regression and logistic regression. The plan was to try more complicated models only in case this classic approach had a poor fit.

The linear regression had as a target feature, the actual number of people present in the room during the survey, which was calculated by multiplying the percentage of people present in the room by the room capacity. The target feature for the multinomial regression, instead, was a categorical feature created by binning the percentage of people counted in the room in 4 levels: Low(0-25%), Mid-Low(25-50%), Mid-High(50-75%), High(75-100%). Together with the Wi-Fi log, we included in the data set the following features, which were selected after conducting the data quality report (Appendix 1 for linear regression, Appendix2 for logistic regression):

- Time, which was considered as categorical variable with 4 levels:early morning(9-11), late morning (11-13), early afternoon (13-15) and late afternoon (15-17). This helped us to explore whether Wi-Fi log accuracy of predicting the occupancy was changing during the day. It is most likely that electronic devices are fully powered early in the morning rather than in the evening. This can potentially result in an overestimation of the number people counted in the room in the morning and an underestimation in the evening.
- Room, which is a categorical variable with 3 levels: B002, B003, B004
- Course level, with 5 levels: Level0 for empty room, Level1, Level 2, Level 3, Level 4,Level 5 for Career talk or meeting. This can indicate us whether electronic devices will be less used during different course levels. First and second level courses are usually more theoretical and they do not require the use of laptops during lectures. Therefore, we might expect fewer connections during this course.

From the data exploration it also resulted that there were not any NaN values in the dataset

and that the Survey occupancy (target feature for the linear model) was not normally distributed, but only assumed those values: 0, 0.25, 0.50, 0.75 and 0.1. We felt that this would have caused an issue in performing the linear regression, but we decided to still keep going into this root as there are statistical tools to deal with this kind of issue. For example, generalise linear model with a Poisson distribution seemed the best solution for dealing with this count data (Zuur et al. 2009)[7]. From the data quality report we detected also outliers for the variables Wi-fi Average clients, Wi-fi Max clients and Survey occupancy, but we did not excluded them from the data set at this stage of the analyses. More details on the exploratory analyses can be found in the Appendix 1 and 2. Once the data quality report was completed, we progressed into the actual analyses, which we are going to discuss in the next two sections. For running the analysis we used R version 3.5.2. We used R, because the most used software used by data miners and data scientists according to the 16th annual KDnuggets Software Pollis [8]. Furthermore, it is a very powerful statistical tool equipped with a variety of packages that perform almost any kind of statistical analyses and that produce high quality graphs. We felt that this was the perfect tool to use in this project, since it was heavily centred on the data analytical side.

3.6.1 Linear Regression

As indicated above, our target feature for the linear regression was the Survey occupancy calculated by multiplying the room capacity by the Percentage of the full room. Since the Survey occupancy was well correlated with both the average number of Wi-Fi logs counted within the hour and with the Maximum Wi-Fi logs measured in the hour (see correlation matrix in Appendix 1), we ran the linear regression analyses with both features. For selecting the features that together with the max logs or with the average logs best predicted the ground truth data, we performed model selection following a similar approach used by James et al. (2013)[9]. We started by running all the models with all the combination of the descriptive features, making sure to always include Average/Max Average Wi-Fi logs for fulfilling our hypothesis. The model run were:

- Survey occupancy ~ 1;
- Survey occupancy ~ Average Wi-fi occupancy;
- Survey occupancy ~ Average Wi-fi occupancy + Room;
- Survey occupancy ~ Average Wi-fi occupancy + Time;
- Survey occupancy ~ Average Wi-fi occupancy + Course Level;
- Survey occupancy ~ Average Wi-fi occupancy + Room + Time;
- Survey occupancy ~ Average Wi-fi occupancy + Room + Course Level;
- Survey occupancy ~ Average Wifi occupancy + Room + Time + Course Level;

The same models were fitted also with the Maximum number of Wi-Fi logs measured in that hour. All these models were run using the 10-fold cross validation (10-fold CV). 10-fold CV was preferred over the validation set approach and the Leave Out Cross Validation (LOOCV), because it is more robust and more accurate in estimating the test error. The Validation set approach splits randomly the dataset into a training set and a test set and it use only the training set for fitting the model. The fitted model is used to predict the responses for the observations in the validation set. For this reason its test error is dependent

on the observations included randomly in the test set and it is usually over estimated. The LOOCV, instead, builds all the possible n-1 training sets, trains the models for all this training sets, which are used to predict the excluded observation. Therefore this technique is computationally expensive and it tends to provide a test error with a high variability, since the folds, used to calculate it, are correlated among each other. The 10-fold CV is similar to the LOOCV, but it only use 10 training set and 10 training set. Therefore it is less computationally demanding and it provide a good estimate of the error. The 10-fold CV was carried out with the package, CVglm. For each model we extracted the overall mean square error(MSE) and we picked as best model the model with the lowest MSE. The best model selected running this analysis on the full dataset had an issue of heterogeneity of variance mainly driven by three outliers. Therefore, we decided to run the analyses on the dataset without the outliers. For more details on this first part of the analysis look at Appendix 1. The analyses run on the dataset without outliers gave better results and therefore, we are going to present them in this document. The results that we got from all the possible models including as a descriptive variable Wi-Fi average logs are summarised by the following table, which shows their MSE.

Model	MSE
Survey occupancy ~ 1	593.14
Survey occupancy ~ Average Wi-fi occupancy	314.55
Survey occupancy ~ Average Wi-fi occupancy + Room	317.93
Survey occupancy ~ Average Wi-fi occupancy + Time	320.69
Survey occupancy ~ Average Wi-fi occupancy + Course Level	330.38
Survey occupancy ~ Average Wi-fi occupancy + Room + Time	323.55
Survey occupancy ~ Average Wi-fi occupancy + Room + Course Level	333.185
Survey occupancy ~ Average Wifi occupancy + Time + Course Level	336.91
Survey occupancy ~ Average Wifi occupancy + Room + Time + Course Level	339.35

The best model was the Survey occupancy ~ Wifi Average logs, with a MSE ~ 315. Similar results were found for the model fitted with the Maximum number of associated client found within the hour. As we can see from the table below, the best model was still the model with only the Wi-Fi Max logs as response variable. However, its MSE was slightly higher than the Wi-Fi Average best model.

Model	MSE
Survey occupancy ~ 1	593.14
Survey occupancy ~ Max Wi-fi occupancy	336.67
Survey occupancy ~ Max Wi-fi occupancy + Room	340.22
Survey occupancy ~ Max Wi-fi occupancy + Time	344.15
Survey occupancy ~ Max Wi-fi occupancy + Course Level	346.84
Survey occupancy ~ Max Wi-fi occupancy + Room + Time	347.14
Survey occupancy ~ Max Wi-fi occupancy + Room + Course Level	349.22
Survey occupancy ~ Max Wifi occupancy + Time + Course Level	354.54
Survey occupancy ~ Max Wifi occupancy + Room + Time + Course Level	356.27

Therefore the Survey occupancy ~ Wifi Average logs was considered as best model overall, indicating that the occupancy was accurately predicted only by the average of the associated

client logs counted within the hour. The other variables did not affect the accuracy of the prediction. Looking at the residuals of the model, the assumptions of linear regression were slightly improved. Even though the plot looking at the residual vs fitted values, clearly showed that the Survey occupancy was assuming more or less the same values. The data were quite normal distributed, and the variance seems more or less homogeneous, but quite similar for a lot of observations. In order to improve the fit of our linear model we tried

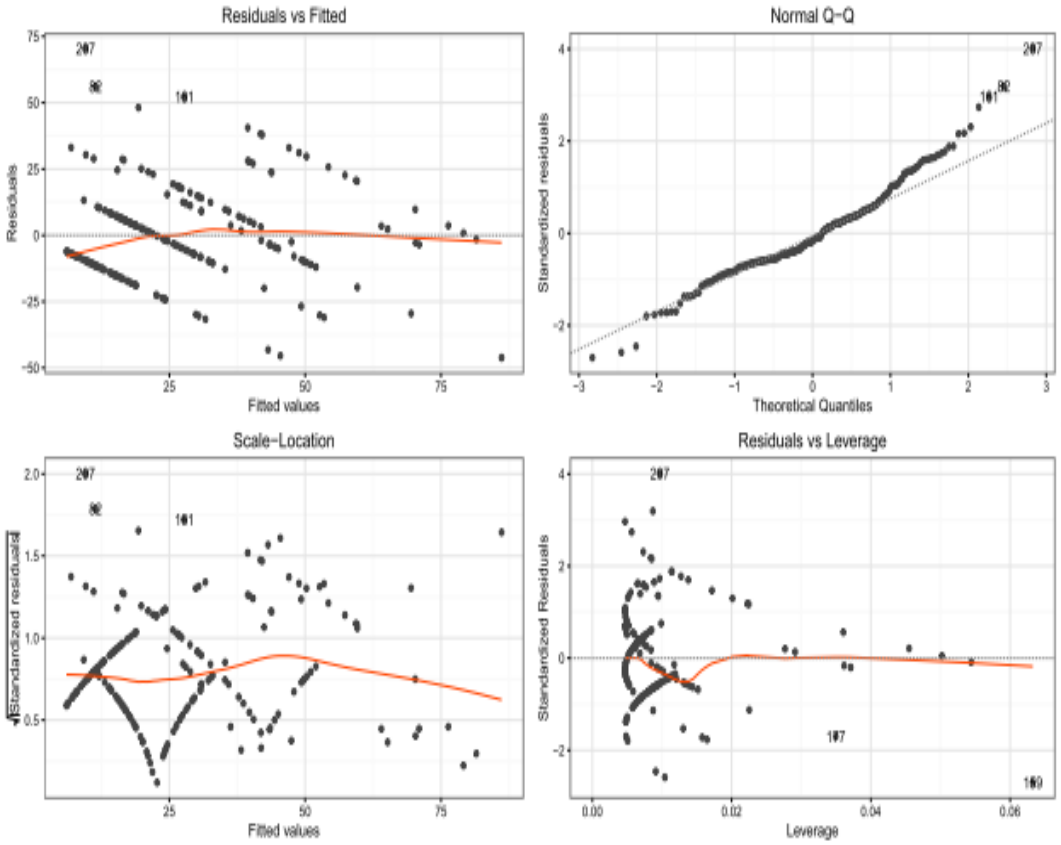


Figure 1: Diagnostic plot for the best selected model:Survey occupancy ~ Wi-fi Average logs

to perform a generalised linear model with a Poisson distribution. The previous analysis demonstrated that this model suffered from overdispersion, thus we corrected the standard errors using a quasi-GLM model, as suggested by Zuur et al. (2009)[7]. We ran all the models using the package glm with family quasi-poisson and cv.glm for running 10-fold cross validation and we took the model with the lowest raw cross-validation estimate of prediction (delta) as best model. From this analyses the best model resulted the Survey occupancy ~ Wifi Average logs with an adjusted cross-validation estimate of prediction error of 390 (Appendix 1). However, when we evaluated the model we discovered that this model was not appropriate for our data and we tried the generalised linear model with a negative binomial distribution. Even this last method was unsuccessful (Appendix 1) and we decided to keep the linear model as our best choice. In order to improve the fit of the linear model we created a mobile application for helping those responsible for the survey to input the exact number of heads counted in a classroom. This will ensure that the occupancy data

will be collected with higher precision and that the target feature will assume other value than 0% for empty room, 25% full, 50% full, 75% full or 100% full. This kind of data are more suited for a classification problem.

3.6.2 Logistic Regression

For running the logistic regression we followed the same procedure that we used for the linear regression. We fitted the model with all the possible combinations of the features by means the 10-fold CV, but this time we used as target feature the Binned occupancy binned in four category (Low, Mid-low, Mid-High, High). For running these models we employed the library “multinom" of the package nnet and preformed 10-fold CV using the package caret. For each model we derived the AIC and the accuracy. The model with the lowest MSE and AIC was considered as our best model. Among all the models fitted with the average of the logs counted within the hour as descriptive variable, the best model in term of accuracy and AIC was Binned occupancy ~ Wi-fi Average + Room with an accuracy of 0.58 and an AIC of 385.62.

Model	Accuracy	AIC
Survey occupancy ~ 1	0.47	477.58
Survey occupancy ~ Average Wi-fi occupancy	0.56	385.62
Survey occupancy ~ Average Wi-fi occupancy + Room	0.58	381.54
Survey occupancy ~ Average Wi-fi occupancy + Time	0.53	393.38
Survey occupancy ~ Average Wi-fi occupancy + Course Level	0.53	393.38
Survey occupancy ~Average Wi-fi occupancy + Room + Time	0.54	388.48
Survey occupancy ~ Average Wi-fi occupancy + Room + Course Level	0.58	401.07
Survey occupancy ~ Average Wifi occupancy + Time + Course Level	0.52	411.05
Survey occupancy ~ Average Wifi occupancy + Room + Time + Course Level	0.55	408

Similarly results were found for the models with the independent variable Wi-Fi Maximum logs, where the best model was Binned occupancy ~ Wi-Fi Maximum logs + Room.

Model	Accuracy	AIC
Survey occupancy ~ 1	0.47	477.58
Survey occupancy ~ Average Wi-fi occupancy	0.51	405.64
Survey occupancy ~ Average Wi-fi occupancy + Room	0.53	399.64
Survey occupancy ~ Average Wi-fi occupancy + Time	0.50	412.65
Survey occupancy ~ Average Wi-fi occupancy + Course Level	0.50	412.65
Survey occupancy ~Average Wi-fi occupancy + Room + Time	0.53	406.12
Survey occupancy ~ Average Wi-fi occupancy + Room + Course Level	0.54	416.19
Survey occupancy ~ Average Wifi occupancy + Time + Course Level	0.51	426.08
Survey occupancy ~ Average Wifi occupancy + Room + Time + Course Level	0.53	422.99

The best model overall was, therefore, Binned occupancy ~ Wi-Fi Average logs + Room model. In the attempt to try to improve the accuracy of the model, we decided to perform the Ordinal Logistic regression, which included the ordinal nature of the target feature in the by defining the probability of an event to and all the all events that are ordered before it.

The Ordinal logistic regression was implemented using the package polr of the library MASS and 10-fold CV was always done with the package caret. The Ordinal Logistic regression did not manage to improve the accuracy of the model (Appendix 2) and our best model will stay: Binned Occupancy ~ Wi-Fi Average logs+Room, which means that the accuracy of the estimate was affected by the room. This model is our main model and it is going to be used for displaying room occupancy.

3.6.3 Analysis output

Once the best models were selected, they were used for creating an R script for running the analyses on the server. The output for these analysis were stored in a data frame. In particular, for the linear regression we stored the predicted values and their bounds (upper and lower limit). While for the multinomial logistic regression we stored the predicted occupancy. The data frame was then sent to the database. The predicted data calculated with the logistic regression were used for displaying the occupancy of the room on the map and the hourly level of occupancy throughout any given day. Instead, the linear prediction were used for interval and heatmap graphs.

3.7 Website - Main information page

The main page was the keystone of the website. It's purpose was to display our data analytics on room occupancy in a clear and concise way. The design and construction was a result of the team's collaborative effort. This section reviews the basic functionality of the page and looks at the more complex features.

The main functionality of the page is to use the internal API's that are generated by the server. According to the day selected by the user the API returns detailed information about modules that occurred and occupancy levels as predicted by the data analytics model. This information is populated onto an interactive and colour-coded map using Javascript. The initial display view is shown below. This is created using Google maps and dynamically loads the icons based on the buildings GPS coordinates.

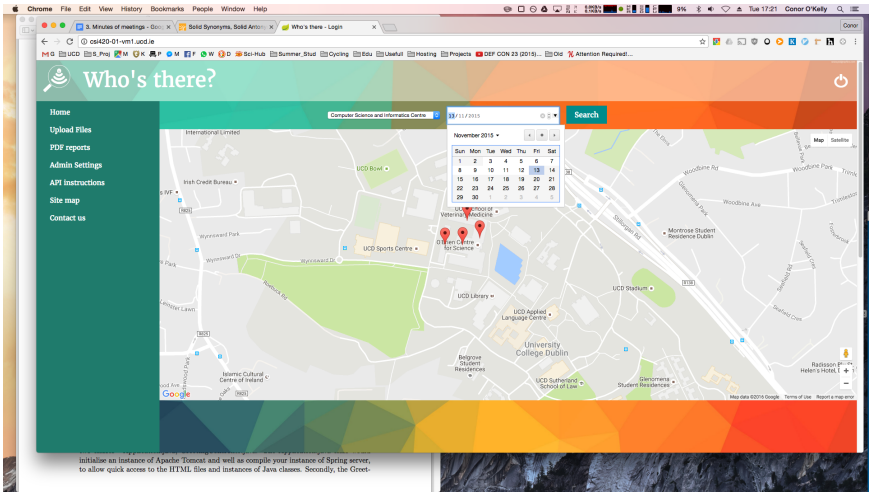


Figure 2: Using the search bar

In order for the user to view information on a specific building and room they have two routes. The first option is to use the search bar at the top of the page. This is set to default a date of 13/11/2015. The user must pick a building form the option list and then pick a date. Currently we have data for the Computer Science building for a two week period. The second option is to click on a specific building icon loading the relevant building page. This will default load the date of 13/11/2015 for the user.

If the user selects a building on a day that no information is currently available for then an error message is displayed. When the user selects the Computer Science building it will load that building and our interactive room display map.

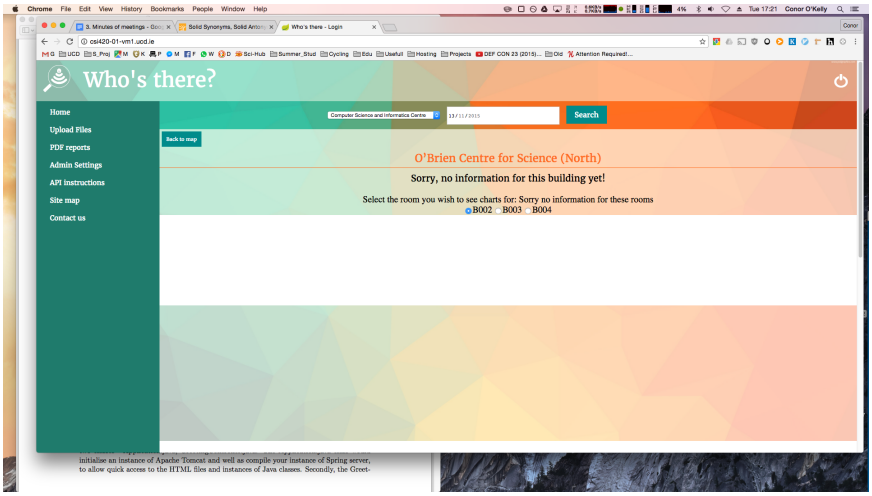


Figure 3: Error message

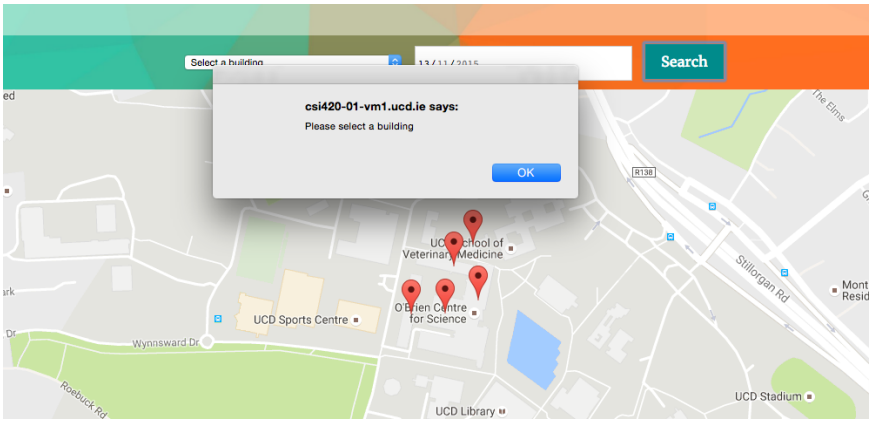


Figure 4: Error message

The team decided on using the library LeafLet in order to generate the final map. This allowed the overlay of a semi transparent layer of colour onto each room. This provides users with high level information about occupancy at a glance. The colour of the overlay is dependent on the results of the logistic regression. The levels of occupancy are indicated by the key located just below it.

The user is able to select different times using the slider or change floors using the radio button at the top of the map. A different date is loaded using the search bar at the top of the page. An example of the finished product is displayed below

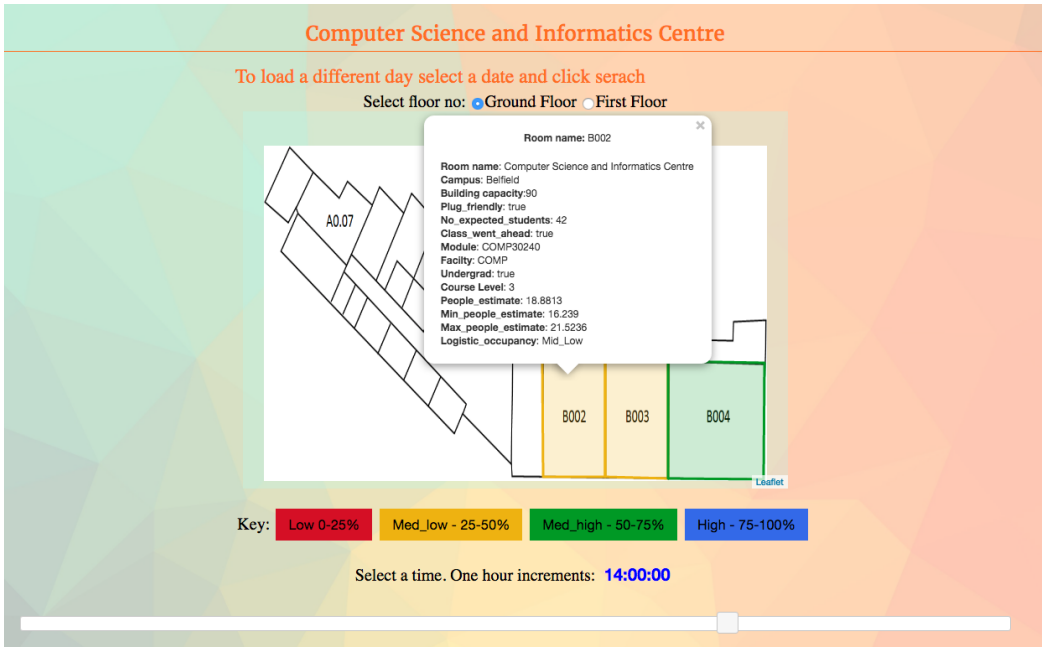


Figure 5: Main information display

The LeafLet library also allows for the creation of pop up windows. In these windows more detailed information is provided to the users about the room. Here we display the remaining information from the API. An example of one of these windows is shown above.

The main purpose of the website being the publication of classroom occupancy information, it was important to ensure that it was displayed in a clear and easy to understand manner. To this end, we presented the data in charts, which represent a good visual support. We used the Google Charts API to generate three graphs on the main page displaying the data associated with a selected classroom at a time and date specified by the user. The first graph is a timeline showing the hourly level of occupancy throughout any given day.

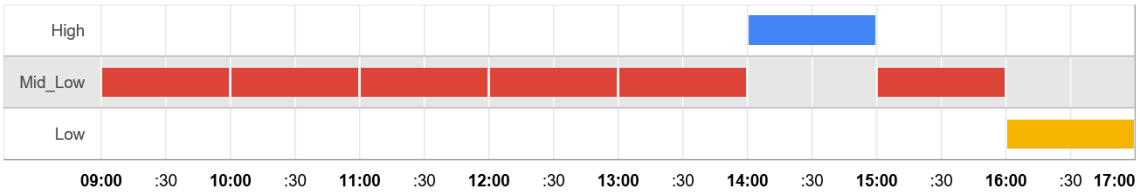


Figure 6: Timeline showing hourly occupancy of a classroom

The second is an interval graph presenting an hourly estimate of room occupancy, with maximum and minimum estimated values.

And finally, the third graph was a calendar with a heatmap displaying the average daily occupancy of a single classroom. At present it only displays data over two weeks, however

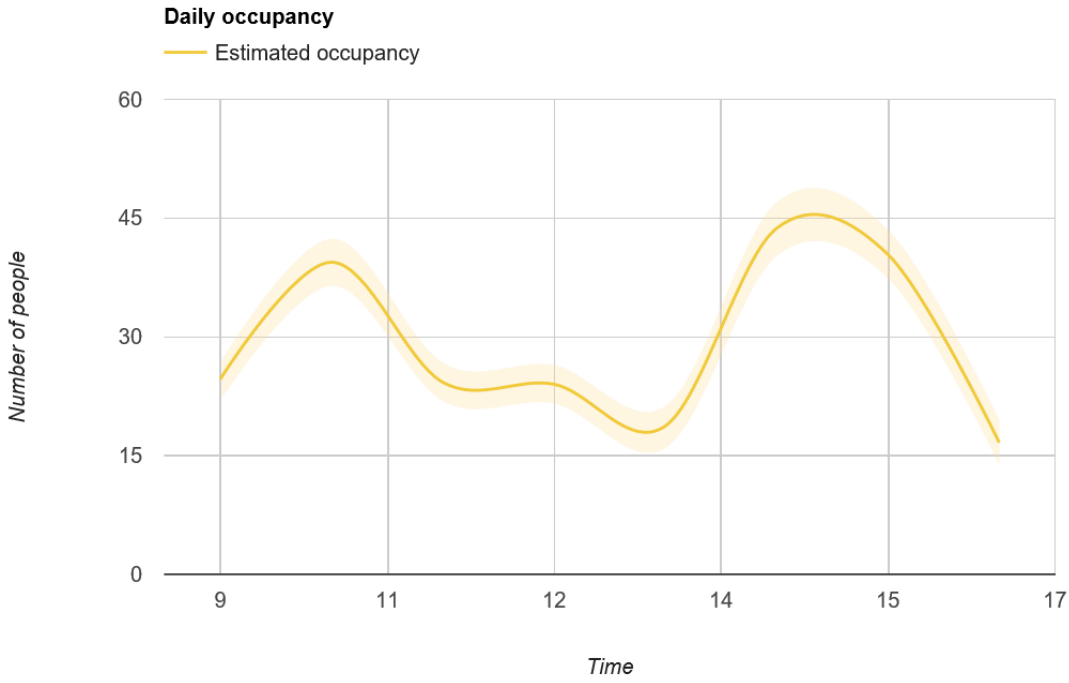


Figure 7: Interval graph showing hourly occupancy estimates of a classroom

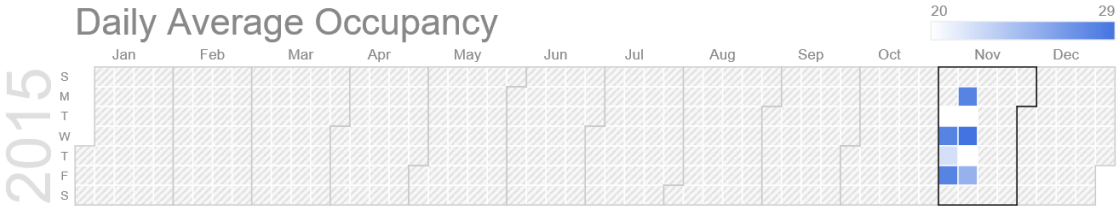


Figure 8: Calendar chart displaying average daily occupancy of a classroom

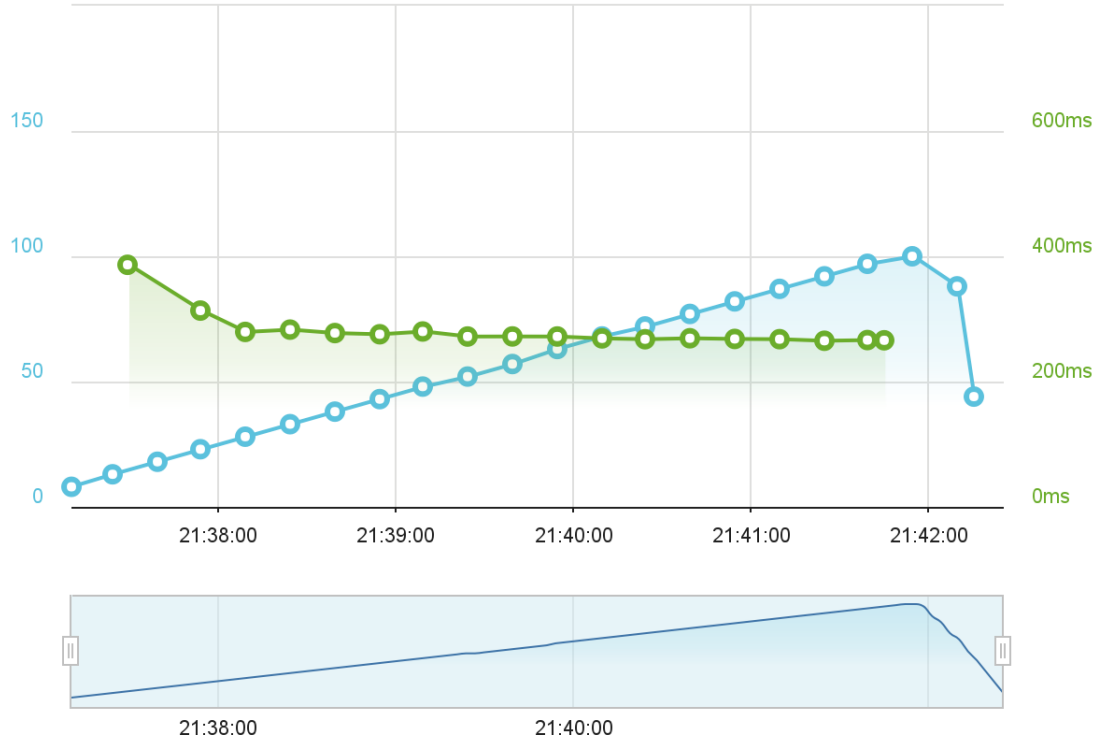
with data spanning over a longer period of time, patterns of occupancy would be easily identifiable on this chart.

3.8 Testing Process/Explaining Code

3.8.1 Load Testing and Performance Issues

As we selected Spring due to its status as a lightweight Framework, we expected it to perform the tasks required quickly. In order to test our assertion that this would be the case, we completed two stages of load testing, the initial load testing - performed prior to Prototype Two utilising Loadimpact as a quick and easy testing method - and final submission tests - performed prior to launch utilising Jmeter to perform more intense tests with more users. During the Prototype Two tests, two pages were examined as test cases - the login page alongside the API. These were preliminary tests to see how well our web-page performs before we had a chance to optimise. During these tests, we found that with a user base

of 100 users connecting concurrently within a period of 5 minutes, the response time was 240 milliseconds when directed towards the login page. The number of users compared to average response time can be tracked on the graph below were users is the blue line and response time is the green.



Alongside this, we performed a similar test against our API, which performed within a similar time-frame of 270 milliseconds within the same test parameters. While we were pleased with the results at the time, we knew we could decrease the time needed for connections. After optimisation, the same test again was performed again, this time utilising Jmeter to allow for its additional configurations in handling Caches, Cookies and processing GET forms. The test performed would have Jmeter navigate to the login page, fill in the form and be redirected to the main page. We found that for the establishment of two HTTP connections, our server gave a response time of 250 milliseconds. While the difference in testing utilities could account for the difference, we are none-the-less pleased with the increased speed, as it can now process two connections in the same time we were previously processing one. However, up to this point, we were only testing the ability to make a HTTP connection, without processing the resources. In order to get a rough idea of actual browsing times, I instructed Jmeter to retrieve all embedded resources, which will process the loading of our Google Maps, the call for our API, alongside fetching the background image. In this test, the results can be viewed below...

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	100	6857	2982	13794	2336.93	0.00%	19.7/min	305.78	955868.2
TOTAL	100	6857	2982	13794	2336.93	0.00%	19.7/min	305.78	955868.2

As shown, an average connection was made within 6857 Milliseconds for fetching the resources belonging to the Login and Main page. While this seems quite slow, it is well within

an acceptable length of time to load the resources belonging to the main.html page, including the google maps API, charts and Jquery which each take roughly 700 milliseconds to load.

3.8.2 Code Review

For the purpose of our project, we planned to implement Unit Tests to the main Java functions. Examples of these functions are the Sql queries, or Runtime.GetRuntime Java methods which assist the general functionality of the Web Application. However, due to issues with time constraints, we did not manage to implement most of these unit tests in the end to assist in the maintenance of our Web Application. Instead, we assured that the results of our code were accurate through Peer Review. Before the implementation of another team member's code into the main server, it would be passed to a different team member. They would attempt to run said code on their local machine, and if successful, it would be pulled to the main server. If unsuccessful, the peer would attempt to find a solution before passing it back to the original designer. This process of passing code between two members of a team assured that each section could be successfully ran on multiple platforms and ensured the positive quality of our product.

3.9 Optimisation Process

3.9.1 Different Languages Implementation

For our project, we utilise the capabilities of three different languages - Java, Python and R. During the initial stages of our project, the team debated on how to include the additional languages into the Java environment. The method which would provide the best results would be to include an embedded version of the language into our central service through libraries in Java such as Jython or RJava. However, this would have required us to include multiple dependencies within the Spring Framework which would allow the embedding of alternative languages, alongside rewriting any existing code we had developed to fit the new structure within the Java environment. As a compromise, we included Java's Runtime.getRuntime().exec() function within the helpers.java class which allows our Framework to efficiently interact with the environment in which it is running - in our case, a version of Ubuntu server - in order to call upon these alternative language scripts via an instance of the command line. The benefits of doing it this way is that we did not need to include any additional dependencies which may bloat the size of our server, as well as increasing development time as we are not required to learn any additional syntax and may run the scripts natively from their relevant file types.

3.9.2 Database Optimisation

As previously discussed, we chose MySql as our database of choice, however, there could still be issues regarding database's speed. As we did not have an opportunity to implement indexed or derived tables, we could not optimise the speed using these techniques and had to rely on more traditional methods. As such, we made sure that our Database is in the third normal form in order to avoid any issues regarding executing SQL queries and any communication with our database handle closes in a quick and efficient way. Additionally, as we noted that a lot of queries to the database from the server would be identical, we felt that any requests to our database may benefit from automated query caching - which

is available for MySql. Enabling this would significantly speed up the website in displaying final results by caching the more utilised requests. While this system may be complemented by the inclusion of the creation of derived tables and indexing to make look up more efficient, we did not pursue that route for the current implementation. However, as the load tests on the API produced a result of 270 milliseconds for one of the more strenuous queries, we feel that the techniques that we have included are satisfactory for boosting database query speed.

3.10 Innovations

3.10.1 API

The API is one of the innovations which we had planned during the initial stages of development. It can be called via the following URL..

`http://csi420-01-vm1.ucd.ie/api/data`

It can then be followed by your query for either Date, Week, Module or Room_no via the following structure...

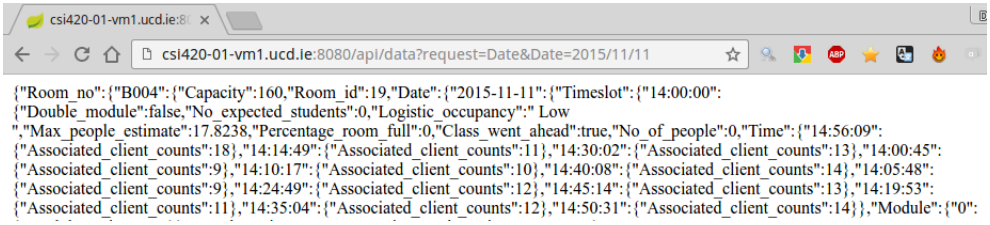
`http://csi420-01-vm1.ucd.ie/api/data?request=[search1]&[search1]=[value]`

It can limit its results by up to two requests by attaching the following language at the end...

`&request2=[search2]&[search2]=[value]`

The API is separate to the security structure of the Web Application and may be called by any user who is interested in our service.

On the technical side, the result is formatted by three major methods within our system. The first is found within GreetingController.java which maps the URL request to a specific script within java. The second, sqlJson, is found within SqlQueries.java which is called by the first which prepares a statement based upon certain search criteria and fetches a resultSet from the query. Finally, a third method, convertJsonFull, which is found within the ResultSetToJson.java class, builds the Json structure based upon certain columns of the return and populates the structure with the remaining columns based upon which table they came from.



While the API can be a useful tool for an external user who wishes to build their own service from the data collected from our Web Application, it is also a useful tool for internal calls, as our main.html page populates its data from our own API service.

3.10.2 Android application

At present, ground truth data is collected manually and provides a rough estimate in ranges of 25% of classroom occupancy. Having more precise figures for a set period of time would be useful to measure the accuracy of the estimations produced by our model and improve it if needed. A tool enabling selected people to easily upload exact occupancy numbers to our database would provide an efficient and accurate way of collecting ground truth data.

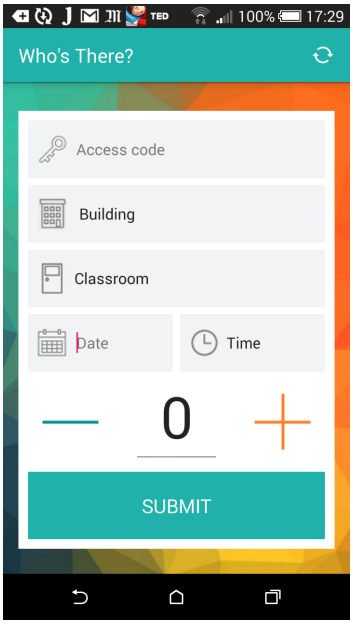


Figure 9: The main activity of the Android application

To this end, we developed a native Android application that allows users to rapidly input the exact number of people in a classroom and send the information to our database. The application consists of a main activity with a form allowing users to select a building, a classroom within the selected building, a date and a timeslot between 8am and 6pm. A text field lets the user input the number of people in the classroom, either by using the soft keyboard or by tapping the plus and minus buttons situated on either side of the occupancy field by respectively increasing and decreasing the occupancy value by 1. As a security measure, we decided to implement an access code feature which only allows selected users who have been given the password to insert occupancy data into our database. After the user fills in all the fields and taps the submit button, another activity opens which confirms that the data has been successfully submitted. When opening the application, a HTTP get request is sent to the server to retrieve JSON classroom data from our API. The data is first used to populate the building drop-down menu, then once a building is selected, to populate the classroom drop-down menu with classroom belonging to the selected building. If an issue occurs preventing the JSON to be accessed, if the server cannot be reached for example, the drop-down menus remain empty. There is a reload button in the Action Bar at the top of the activity which sends the same get request to the server and populates the drop-down menus if the request was successful.

The android application was built using the Spring library for android to facilitate HTTP requests to our server-side Spring application. The use of Spring on the server and on the Android application greatly simplified the HTTP request process. The Spring library for Android provides an Android version of the RestTemplate module, which is a java-based REST client. It uses the HttpMessageConverter interface to marshal objects into HTTP request body and unmarshal responses into objects [10]. In our case, we used the MappingJackson2HttpMessageConverter class which implements the HttpMessageConverter interface and can read and write JSON. The called API returns an array of classrooms including, among other information, classroom ID and number, campus and building. The message converter unmarshals the JSON into an array of Room objects. From this array, an array of Building objects is derived and each room is then associated with the building it belongs to. The Room class uses Jackson annotations to correctly map JSON values to class variables. The **@JsonProperty()** annotation, located above variable declarations, associates JSON keys with the variables. And the **@JsonIgnoreProperties(ignoreUnknown = true)** annotation inserted before the class definition prevents errors occurring due to unmapped JSON keys. This is useful if the JSON structure changes and keys are added or when the JSON

string contains a number of values that are not needed. This is an example of the two annotations:

```
@JsonIgnoreProperties(ignoreUnknown = true)
public class Room

@JsonProperty("Building")
private String building;
```

When submitting the form, a GroundTruthData object is created with variables for each of the values that the user input in the form (access code, building name, room number, date, time and occupancy number). A HTTP post request is made using RestTemplate and the GroundTruthData object is marshaled into the HTTP body request by the message converter. Both HTTP requests time out after a period of 3 seconds. On the server side, a controller was added to the Spring application which intercepts the request and unmarshals the JSON into a GroundTruthData object. The controller checks the validity of the access code. For the purpose of this project, the access code is hard-coded in the controller. If the access code provided is invalid, the controller generates an unauthorized (401) response, otherwise it returns a 200 OK status code. A query is made to the database to insert a new row in the Ground_truth_data table with the values of the variables associated with the GroundTruthData object.

The HTTP requests were handled within separate threads so as not to block the activity going on the main thread, also called the UI thread. All system callbacks are handled on the UI thread, so running the requests on the same thread would block all interaction with the user interface for the duration of the request. In order to perform the two background operations on different threads, we extended AsyncTask in two subclasses within the main activity. AsyncTask is a class which enables operations to be performed asynchronously and allows the results of the operations to be printed on the main thread. The doInBackGround(Params...) method must be overridden as it is the one that performs operations on the background thread. It is in this method that we make the HTTP requests. We also override the onPreExecute() method in which we open a progress bar on the UI, and the onPostExecute() method in which we close the progress bar and process the result of the HTTP requests.

The Android application was designed to be straightforward and easy to use with a minimum of items on the screen. We organised the form in logical order, with access code first, followed by location and time information and finally room occupancy. We used icons in the input fields to provide quick visual indications on how to fill the form. Messages appearing in toasts (or small pop-ups) inform the user that an error occurred when the server cannot be reached or returns an unauthorised status code, or display a warning message when the form is incomplete on submitting. These messages guide the user and promotes a smoother user flow.

3.10.3 UX Design

User experience (UX) can be defined as a person's perception when interacting with a specific design. A product may feature all needed functionalities but may generate bad user experience if its design is not intuitive, leading users to spend a lot of time understanding how the product works. Bad user experience may also occur if a product takes too long to complete an operation because of the number of steps required from the user for example. User-centered design focuses on how a product works on the outside, ignoring its inner workings. We used principles of UX designs while preparing the user interfaces of the website and the Android application.

We prepared personas, which are fictitious characters representing the different types of expected users of a product being developed. They serve as examples of users, defining their goals and behaviours, and help designers and developers better understand the way a real user would interact with a product. We created three personas to represent the three types of users we defined in our SRS document: an administrator, a professor and a student user. We gave each persona different levels of computer proficiency, defined their goals and described how the product could benefit them.

During the first sprint of the project, we drew paper prototypes for the web and Android applications. Drawing the prototypes was useful to refine our initial ideas and identify issues with them. We then built a wireframe version of the prototypes using cacoo.com, which allows users to share wireframes and other diagrams. This gave us all the possibility to make changes to the prototypes and produce cleaner templates to base the design of our user interfaces on.

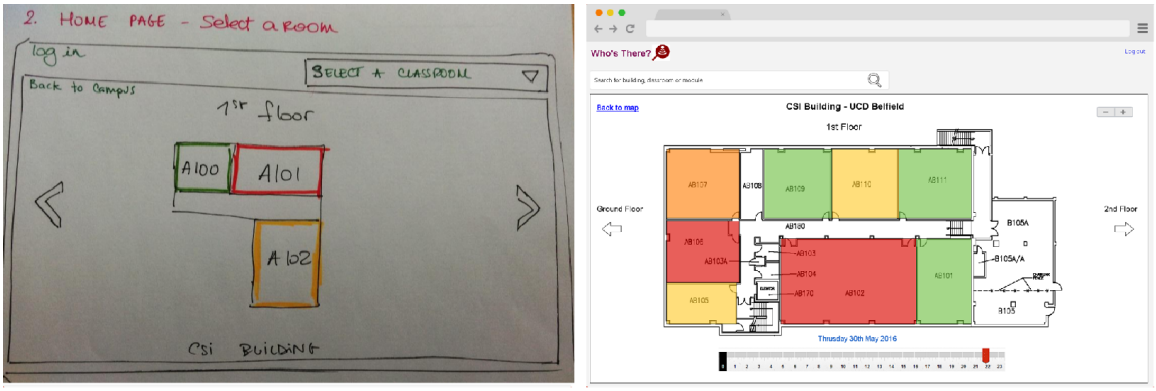


Figure 10: Examples of paper and wireframe prototypes showing the main page of the website

A style guide was also set up on frontify.com (and can be viewed at `https://app.frontify.com/d/nZP2dQ8rYb0V/who-s-there-style-guide`. It contains styling standards for designing the web and mobile front-ends. It was used as a reference and ensured that any changes or additions we made to the user interfaces were done in a cohesive manner. The guide includes the set of colours that can be used, with the RGB and hexadecimal values. It also includes typography rules, with font families, size and colours to be used and the associated CSS code. Finally logos, icons and images are downloadable from the guide.

During our final sprint, we prepared and organised a usability test. Usability testing is a technique used to evaluate the design of a product by observing users while they complete a

set of tasks on the application. We selected three potential users, all current UCD students, and asked them to complete the test. The usability test was conducted using a live version of the website and Android application. A laptop using Camtasia recorded the on-screen activity and the participants' comments. Two mobile devices were used to test the Android application: one on which the participants tested the application and the other recording the participants' actions and comments. Participants were asked to sign a consent form before the beginning of the evaluation. They were then given a first task scenario for the website and asked to read it out loud. The participants were then asked to complete the tasks on the website while being recorded. After completion of the tasks, they were asked to fill in an on-line survey containing a set of 20 questions to rate the efficiency, control, learnability, helpfulness of the website and their emotional response to using the website. The participants were then asked to repeat the same steps, with a different task scenario, on the Android application. The results of the usability test were used to write a set of recommendations on how to improve the usability of web and mobile applications. The main issues discovered during the tests were linked to the lack of intuitiveness of the slide bar and graphs on the building page of the website. On the Android application, the main issue was the occupancy field in the form, with participants using the plus and minus buttons to increase or decrease the number without realising that they could input a number using the soft keyboard. For more information on the tests and their results, please refer to the Usability Test Report in the appendix.

3.10.4 PDF report

Similarly to the development of the API, the idea of creating PDF reports was introduced in the early stage of our project, but it was only implemented in the final week when the rest of the application was fully working.

The idea behind the creation of the PDF report was to summarise the daily occupancy of the room in a unique PDF page with graphs and table, which could be immediately by UCD administrative personnel without the need of navigating throughout the website. We decided to create these documents using R and its packages: knitr and rmarkdown. The beauty of these packages is that R code can be embedded into the markdown text for running statistic or producing graphs, the output of which are displayed when the document is rendered it into a PDF file. The PDF file is accessible by simply completing a form, where the user is required to select a building, a classroom, a date and press the submit button. The selection of the building is simply needed for populating the classroom drop down menu for showing only the rooms belonging to that particular building. Once the form is submitted the two input variables, Classroom and Date, are stored in a object and an HTTP post request is made. As a consequence the created object is injected into a R script, which in turn triggers the creation of the PDF document by inputting the two values in an a R markdown file (.Rmd extension). The two input values, classroom and date, are used for building the URL for requesting the API containing the occupancy of the selected room in the selected date to the server.

From the API request a JSON file is returned and it is parsed using the R package into a data frame, from which values and key can be extracted and used for building the table and the graphs for displaying the information requested. Once the PDF is generated is saved in a folder in the server. At this point the server will make the PDF automatically download into the browser windows. In order to avoid any conflict among users requesting the PDF at

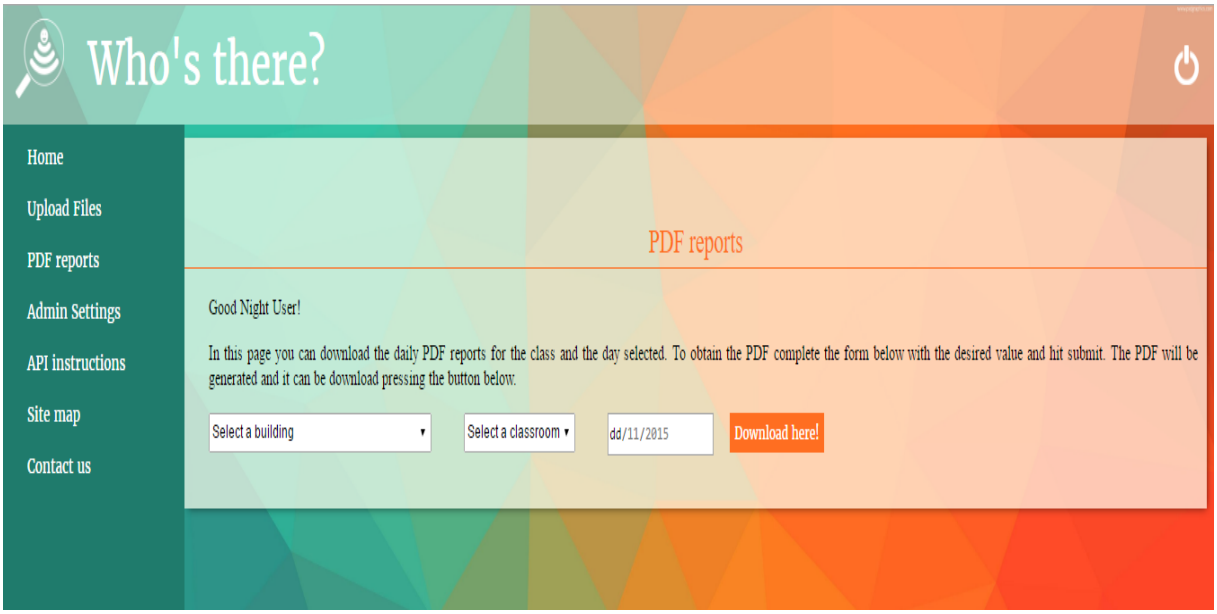


Figure 11: Form for requesting the PDF report

the same time, the name of the PDF report is randomly generated. The PDF will then be deleted from the folder to avoid filling the server with unwanted information. The process just described can be summarised by the following picture. At the present only a daily PDF

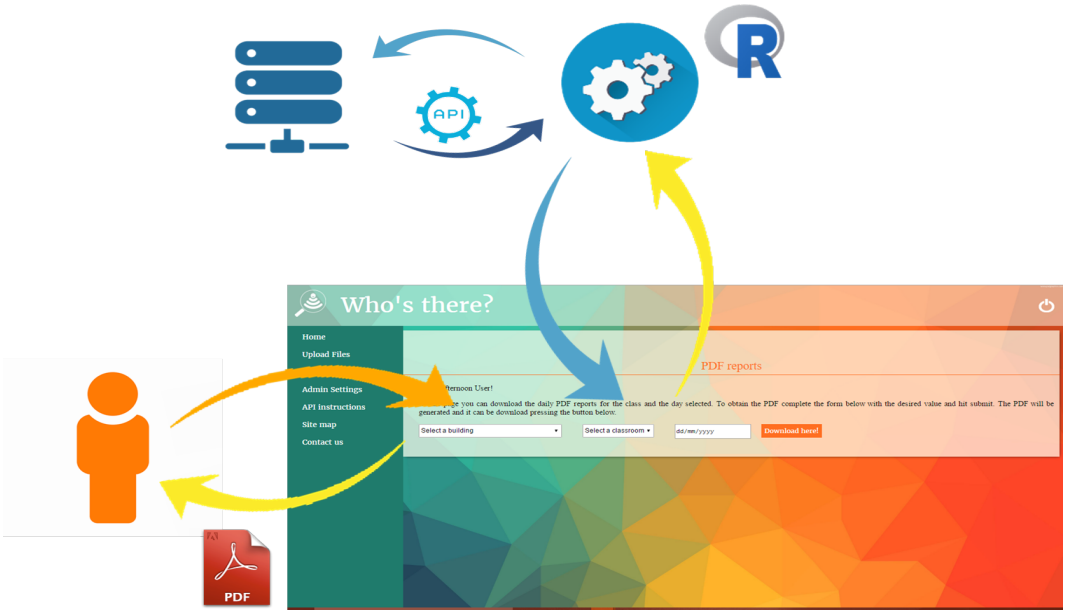


Figure 12: Form for requesting the PDF report

report is available for a single classroom, but another script can be implemented at future date for generating weekly PDF report for each class and building.

4 Personal Contribution

4.1 Overview

The penultimate section of this report looks in detail at my personal contribution to the team's overall project. Throughout the project I have created detailed minutes and weekly blogs detailing my progress. From this I draw the majority of the content for this section.

My personal contribution can be divided into two main areas. Firstly, tasks I have completed throughout the project and secondly my role as team coordinator in guiding the overall development process. It would be wrong to take sole credit for the success of our project. Instead I will review my personal work and the structure I attempted to provide for my team to excel.

My first task as team coordinator was to help set ground rules that would dictate how the team would work together. Reestablishing the roles within the team from project guidelines allowed each member of the team to take ownership for their section. As team coordinator I decided that my main purpose would not be to take charge of the team but instead my objective was to facilitate the creation of a strong and cohesive unit. This would allow all the team members to be on the same page while working towards a common goal with room to disagree on how we implement the project.

My second main contribution to the project was the establishment an environment of shared tools. This would facilitate communication and help to focus the team. The collaborative tools were as follows; Github for code, Backlogs and Trello to plan and track task progress, WhatsApp and Slack for communication between the team.

One of my most important additions to the team was the effective management and record taking during our meetings. As coordinator I ensured that I took clear and concise minutes during all of our meetings. Our meetings and minutes would take the following format. First we would lay out an agenda of high level topics to be discussed and act as a guide for the meeting. During the meeting I would ensure to take down all relevant information. At the end of each meeting the team would agree upon a general conclusion. This would outline the areas of individual focus for the immediate future.

The meetings were our version of an effective stand up meeting. It allowed each member to outline their current progress and set goals for the next few days. It also facilitated members of the group asking each other for help when working on difficult sections. The minutes and backlogs allowed the team to create a written record of what they had committed to and also provided guidance. As team coordinator I used backlogs and our minutes conclusions to direct progress via roadmap goals. The full minutes are attached in a separate document.

One of my first personal tasks was to help with the creation of the software requirements specification (SRS). The group decided in order to properly plan the project and develop the roadmap we first needed to agree upon its core requirements and functions. This would allow us to build a concise vision of the project that we could verify against customer feedback. My main focus for this was on the data input and storage requirements. These were two areas I would continue to focus on for the rest of the project. Putting time into creating

a concise vision for the project allowed for the team to have a clear goal for the finished product. This helped to keep all members on the same page throughout the process.

My next main contribution was the creation of the database schema and selection of database technology. As a group we decided to research possible database options. My main focus of research was SQLite, MongoDB and MySQL. As a team we settled on MySQL for reasons outlined in previous sections of this report.

The next associated job was proposing a rough draft of what our database schema should look like. I personally thought correctly establishing this core feature was crucially important to avoid expensive rework or recoding in the future. I began by focusing on looking at our data sources and requirements outlined in the SRS. I then brought this forward to a group review. Working together my team helped me to flesh out the rest of the schema.

The final stage for our initial draft was to work through the normal forms. Eventually we achieved a third normal form on our database ensuring organisation while avoiding redundancy and related anomalies. This extra effort allowed our first schema draft to survive without significant change from the start to the finish of the process. A key exception to this was including data analysis results into the main section.

This task logically lead on to the creation of a system that would populate the database from the raw information sources. In this task I took the primary lead and was assisted in two modules by Devin. My overall goal was to design a system that would be able to handle data inputs in a very modular way. Before I started coding I created an overall architecture to follow.

In this project there were three main data sources; timetables, ground truth data and wifi logs. Each data source has it's own module that correctly extracts the information from the raw file and returns it as an array of dictionaries. These dictionaries take a standardised format. The dictionaries are then returned to a single controller module that handles the input of the data into the database. The controller module also handles secondary tasks. Examples include ensuring the database exists, creating the database if required or populating tables with derived or additional information.

The motivation behind this structure was to allow modules to be easily replaced in order to adapt to new data input formats. Throughout my development of the data input system I tried to put this into practice. Functions would be as small as possible, have specific tasks and be well commented. This should also allow for an easier debugging process.

The most difficult part of this overall process was finding the correct library to deal with Microsoft excel files. After looking at numerous libraries I settled on using Openpyxl. One feature that seemed to be missing from a number of libraries was an efficient method of handling merged cells. Only the first cell in the merged range would hold a value and the rest would be blank. I ended up writing a function that would 'un-merge' identified ranges and copy the data from the first cell into the rest of the range. I hope in the future to refine this function and submit it to the development process to be included in future versions of Openpyxl.

During the construction of the data input module I realised it would be important for users or admins to gain feedback on the success of inputting a particular file. The controller

module has a basic level of error handling that created success and error message for each file that is processed. This information is then input into the data input log table of the database. This is a feature we intend to implement in a future version of the website.

4.2 Spring server and upload page

During the overall development process I tried to keep up to date on the core features the team was working on. This enabled me to have a good understanding of the technologies as the project progressed. A core area that I had no previous experience of working with was the Spring server. As such I arranged that one of my next tasks would be the development of the upload page. This was closely tied to the data input programming work.

Creating the HTML and CSS for the upload page was straight forward. Connecting it and making it work with the Spring server was a lot more difficult. At the start of this I realised that I had limited experience using Spring. My first task was to create a new Hello World Spring website in order to grasp the basics. Thankfully there are numerous tutorials online to follow.

This combination of starting from scratch provided a wealth of community information which allowed me to develop the page and gain a much better understanding of the Spring server. Working together with Devin we connected the upload process with the data phrasing and data analytics script. This meant that once a file was uploaded it would be automatically input into the DB and the data analytics process would be activated.

4.3 Main website page

My final main task for the group was turning the data generated by our internal API into an aesthetically pleasing visual for our customers. As a group, we decided early on before proper investigation that we would create a map for each floor with colour overlays. Each floor of every building would have an overlaid semi-transparent colour. This colour overlay would provide an indication of room occupancy at a glance. When clicking on a room it should then display information about that room.

These features turned out to be significantly more difficult than first anticipated. The code required had to be implemented in Javascript on a page that already had lots of unstructured code. As a group we had not decided on a formal style and format for our Javascript and this turned out to be an issue.

After significant research on the web I found the Leaflet library that allowed me to implement a solution. Even after deciding on the correct library it still took two days to flesh out each area of functionality before I could begin to integrate my code into the page. The main page already had a significant number of features implemented. These included; Google maps with data being dynamically added, hidden divs for different buildings and a search bar.

The first step was to move the majority of the Javascript off the main page into a separate file, this could then be imported on load. I followed this up by aiming to encapsulate as much of the code as possible into separate functions that could be called upon when required.

After this it was a relativity easy although tedious process to implement the actual map, overlays and popup divs. As with any coding I do I tried to ensure this was done as dynamically as possible. This is done by ensuring that most values and colour are generated on the fly rather than hard doing in. Getting the search bar and other features to work required me to add error catching to the page. This was achieved by using a simple check to make sure variables were not blank on search submission and if they were then notifying the user with a simple alert. The finished results are shown in the appendix.

4.4 Other website work

Towards the end of the project I also helped out with some basic website work. This was more polishing other members work or adding small pages such as the error page. I also did some minor refactoring of code to increase readability and page mapping in the server. This helped to finished off some of the more minor pages that added background to our site.

4.5 Work conclusion

Even though I felt that I had contributed a lot to the project I feel it is important to point out there were many areas that I was not involved in. Our software project was truly a team effort. Each member of our team played a crucial and equal role in the project.

4.6 Role as team coordinator

It is important that I finish this section up by reviewing the contribution I felt I brought to the team as the coordinator. At the start of the project this was not my ideal role choice but one I grew to enjoy. Often software engineering projects are not let down by their lack of technical skills but instead by their management process.

From the start it was important to have a roadmap that was both achievable and tangible. Coupled with a creative environment with high levels of team trust and focus that would allow us to achieve our goals. During each of our meetings I aimed to keep them short and focused on the task at hand. By writing a conclusion to each meeting it helped to create an easy reference point on which each member could look back upon. As members in my group mentioned this proved extremely helpful in keeping on target over the three months of the project.

One of the most important features of our group was being encouraged to discuss issues that we were experiencing. I tried to take this lead by ensuring that any problems I ran into, I shared with the group. Sharing problems without judgement allowed the team to work together and for each member to excel in their strengths while helping others to further develop their own skill set. Establishing an environment for this allowed minor issues to be dealt with early on before they become major ones.

A secondary role to this was dealing with team issues as they arose. One of the first issues that the team encountered was when individuals felt the work was not being divided evenly.

I was able to deal with this by ensuring from then on that more discussion took place before the allocation of work.

A major part of being team coordinator was ensuring that the team had a vision of what the final product would look like. As mentioned previously I helped to facilitate the creation of this vision by ensuring our team spent adequate time on the SRS.

4.7 Conclusion

Overall I effectively fulfilled my role as team coordinator. I think one of the most important aspects of this role was to realise that I was not to be in charge of the team but instead the goal was to facilitate the creation and working environment of a strong cohesive team. By working efficiently and together as a team we were able to meet all of the original project goals.

5 Future work

The majority of the original features that the team established in the SRS at the start of the project were successfully implemented. A number of features were also discussed at the start of the project but identified as beyond the scope in the current timeline. This list of future features grew as the project progressed. In this section I will overview possible future work.

An issue that was identified during the project was long return times from the SQL database. Complex requests joining many tables could often take a few seconds to return. This in turn would affect load times of the webpage as it waited for this data. Two possible solutions were identified, optimisation of the database and caching of frequent requests.

The first solution would require the implementation of index and derived tables for frequent request in the database. This should allow requests to be returned significantly faster.

A specific API call is made each time the main page loads. This call should be cached at the server level instead of being pulled each time from the database. Server side caching will also be rolled out to other sections of server where loads times would benefit. This could also be combined with local level caching in the user's browser.

The registration process for our website currently does not require the user to have an activate email address. In the future users will be required to sign up and then verify their email using a generated link.

One of the innovations in the project was the use of a UX design process. In the future this can be expanded upon by doing more surveys and usability testing. The feedback from these interactions will be used to improve the user friendliness of the website and overall design.

Due to the small data set available in this project the team was restricted in the type of data models it was possible to implement. As the size and number of distinct features of the data set increases the type of data modelling possible will expand. The group propose to explore the use of random forest and Lasso regression models. This would be combined with new information as the team explores the use of OS fingerprinting to identify the type of device connected to an access point.

An additional feature identified for the upload page would be to provide feedback for the users as their files are input into the database. This feature was implemented in the data uploader but the results still need to be displayed on the website for the users.

A final future goal for the team is to improve the coverage of unit testing across the code base.

6 Conclusion

Overall the research practicum assignment was a large success. Through effective project management and hard work the team achieved all of the core requirements and four separate and significant innovations. To conclude I will identify the main features that make our project unique and summarise the best sections.

The overall project stack was as follows. The whole system was run from an Ubuntu server. To provide our web-server, we used a combination of Apache2 and Apache Tomcat. The functionally of the server was implemented in Java using the Spring library. We then used MySQL for our database, Python for managing the data and R for data analysis.

This combination of technologies allowed the team to produce a project with a number of advantageous qualities.

- The software functioned consistently under load testing.
- A modular design that is easy to update and adapt to new environments.
- A choice of libraries that made it easy to extend.
- An efficient overall product that would scale to much larger requirements.

In the future it would be easy to add new rooms, building and data models to the finished product without requiring large changes to the base software.

Beyond the core requirements the team was able to successfully implement four different innovations. The most impressive of these was the creation and deployment of a full Android application. This facilitated the collection of live ground truth data in an easy to use way. Next was the use of UX design process, in order to maximise usability and design aesthetics of the website. Another important feature of the website was the ability to download customisable PDF reports that allowed users to easily share information. The final innovation was the creation of external APIs allowing our user full access to our data set.

If we were to start the process again there is one of the main changes I would suggest. The establishment of consistent coding practices and training each member to correctly use collaboration tools would have been invaluable. Once the development process has started these behaviours can be difficult to change. In future each member should train on the best practices for using GitHub and improve the way they layout their code.

At the start of this development process our team invested significant effort into the creation of our SRS and related documentation. I found this creation of a roadmap and a specific vision to be invaluable in guiding the team to a successful outcome. As such we have also included all our additional documentation in our code folder in the project documentation directory.

The creation of any large scale software is the result of a collaborative effort. Our team worked together consistently throughout the development cycles of this project. Each sprint brought us a step closer to the end goal. As a final note I would like to thank my team

for working with me to finish this project together. It was an enjoyable and educational experience. The software developed here lays good groundwork for future development to roll out to a much large scale implementation. It's a project we can all be truly proud to have taken part in.

7 Appendix

7.1 Overview

The appendix of this report contains my profile, class blog and personal blogs. The extra documentation created by the team is included in the additional documentation directory in our code folder.

These documents were originally planned to be included in the appendix but this brought the report to over 200 pages so have been left out.

7.2 Portfolio

7.2.1 My Profile

My name is Conor O'Kelly and I previously completed my undergraduate degree in Economics and Business In Trinity College Dublin, Ireland (TCD). I have always had a big interest in computers, while learning about Econometrics I gained my first exposure to using programming languages. This encouraged me to explore the use of Python for more simple tasks and eventually lead me to consider computer science as a full time career. I felt that a solid ground in business skill and economics would be complemented by strong technical skills in the IT industry. The ability to work in and manage teams successfully would carry over from my previous experience.

My Master's in UCD allowed me to develop fundamental skills in computer science, that makes it possible to self-learn new topics. I found that a significant amount of time is always spent learning in order to develop skills required to work with new software and libraries. The course has shown me that it is not current knowledge that is crucial but instead the ability to learn new tools and techniques.

I have also worked over the period of the course to supplement my core skills by learning how to use GitHub and familiarising myself with working with the Linux environment.

7.2.2 Learning to date

During the first semester of this master course there was a strong focus on learning the fundamentals of computer science. The development of a deep understanding would allow me approach high level problems with good insight. The courses Computer Architecture and Operating Systems taught me about the central components that an Operating Systems comprised of and how these work together. This was complemented by an education in how networks are created. In this class we explored basic error catching and correction also received a high level overview on the theory of networked systems. I also became familiar with the use and creation of relational databases (SQL), creating queries using relational algebra and achieving the normal forms.

A core aspect of computer science is a good understanding in object oriented programming. During the first term we explored this in Python before then moving to Java to become familiar with both languages. We looked at the advanced features of designing and building object oriented software.

In the second semester we continued to develop this skill set. I worked individually to data mine and then analyse a data set learning the basics of Data analytics. First I did exploratory analysis on the data set before cleaning the data and attempting to model relationships. In web development we learned how to combine HTML, CSS, JavaScript and PHP into a functional website. In Data structures and algorithms we learning about methods for algorithm design while exploring existing examples. Finally a key module of the semester was Software engineering where we learned as teams the processes required to plan and then manage the development process of a software project. Here we refined planning skills by creating a software requirements specification (SRS) and managing development using an Agile methodology known as scrum.

TEAM: WHO'S THERE

ESTIMATING ROOM OCCUPANCY USING Wi-Fi LOGS

Ophélie Alliaume, Conor O'Kelly, Silvia Saloni & Devin Stacey

INTRODUCTION

Wi-Fi technology is ideal for monitoring room occupancy, given the low cost and ease of installation. Other alternatives, like survey, are expensive and time consuming. UCD is spending 25,000 euro each year for conducting a survey with the hope of optimising the classroom usage.

AIM

Use Wi-Fi logs data for predicting room occupancy of UCD campus and displaying it on a web application to cut the cost of the survey.

METHODOLOGY

- Wi-Fi logs, Timetable and Ground truth data were cleaned and upload into a SQL database (DB), using Python script.
- Spring server responsible of interconnecting all the components of the system
- API generated by the server for external and internal use.
- Use of UX design for making the application user friendly

ANALYSIS

- Occupancy was estimated with 60% accuracy using multinomial linear regression .
- Target feature: % of the full room full binned in: Low (0-25%), Mid-Low (25-50%), Mid-High (50-75%) and High (75-100%) ;
- dependent variables: Average of the Wi-Fi logs measured in 1 hour and Room.

RESULTS

- Data were loaded to the DB without losing any information and duplicates;
- Server handling login system (for restricting access to non staff and administration) and contact us e-mail.
- Web application displaying the estimated room occupancy on a map;
- API used for generating graphs and PDF reports that can be download by user;
- A mobile application was developed for inputting ground truth data for correcting the prediction of our model

CONCLUSION

Our system has successfully covered the requirements of the project and it can be used to facilitate UCD staff in optimising room usage and course timetables.

SCHOOL OF COMPUTER SCIENCE
University College Of Dublin
Belfield, Dublin 4, Ireland

7.2.3 Project role as coordinator

My main role for the research practicum was to act as the coordination leader for the team. It was my job to manage the team's development process. Ensuring the team first laid the solid plan and then meet objectives throughout the development cycle. I would solve team problems and provide a framework of shared tools for collaboration. Aside from this it was also my responsibility to manage the team meetings and facilitate discussion.

One the most important lessons learned from this role is that coordination is most effective when the team is encouraged to coordinate itself. This requires that each individual has a clear understanding of the end goals and current part they are to play. I provided focus with a combination of long term goals and daily tasks enabling the team to work consistently over period of three months.

As coordinator I ensured that I took clear and concise minutes during all of our meetings. At the end of each meeting the team would agree upon a general conclusion. This would outline the areas of individual focus for the immediate future providing focus for each member. Secondary to this was creating task backlogs where each member had their task for the current sprint documentated.

These elements of project management allowed our team to work together as a strong cohesive unit. We were able to meet the core requirements and our additional requirements. Being the project coordinator gave an important education in how to form a strong group and work through problems as a team. This in itself provided fantastic real world skills.

7.2.4 Main interests in Computer science

My main areas of interest in computer science is software development and security. During the term I have used my skills to create small extra projects. One example I did for fun as using a Raspberry Pi to create my own in home media server. This gave me experience in researching as task and following it to completion. During the project I found a library I was using to deal with Excel files to be functionally lacking. It forced me to write my code to achieve the required results. I plan to format the code, add tests, comments and hopefully get it included in a future version of the library. This will be my first experience with contributing to an open source project.

Security is currently very topical in the IT industry and an area that I find quite interesting. I hope to be able to improve my skills in this area of the next term to pursue work in the field. I discovered a real security issues early this year when testing an application for SQL injection attacks. I found this issue when using Subways loyalty card system which uses QR codes. I created a new QR code with SQL statement inserted to test their system.

7.2.5 Questions asked on Wednesday sessions

- How to achieve a balance between modular code and finishing work
- Asking another team about their methods used to develop their website map to compare it to my own.
- The best way to create specifications for a project.

  Working top down vs bottom up.
- How best to address the security concerns of the customer

7.3 Class innovation blog

7.3.1 Intro

Over the course of our semester long project I have learned a lot about working together as a team and on building software. One of the most important lessons I have learned is the difficulty associated with providing accurate estimations for time to complete work.

When dealing with technologies that the team are unfamiliar with It is very difficult to estimate complexity. As happened in my team this can result in a situation where we committed to the outcome of creating color overlays for our map. We decided this would be ascetically pleasing and give users a good high level overview of room occupancy.

The problem arose when it came to implementing these features, the group had done virtually no research on how it should be done. We were unfamiliar with an existing solution and unsure which library might be suitable. This made it next to impossible to estimating how difficult the task might be or how long it might take.

This example highlights a common problem I am sure the majority of groups experienced as they began to use new languages and libraries. An important tool I discovered as part of the sprint process was the use of spike investigation. This is a special type of story or task that is allocated in the backlog.

There are two main types of spike investigations technical and functional. A functional spike investigation is used to to analyse how functional behavior will be implemented in the code base of a project. It identifies how an overall task my be broken down and its areas of complexity and risk. A technical spike is used to determine feasibility and impact of design strategies. [http://www.scaledagileframework.com/spikes/]

In short a spike investigation is used to provide information about an unknown technology or feature before the team commits to a process or to implementing it at all.

7.3.2 Main section

In this blog I outline their use and give examples from my personal experience on their benefits. As stated earlier the main area of concern of my team was the implementing of the interactive color overlay upon our custom map images. In this example I follow a functional spike as I explore implementing the feature.

The first issue was of course the selection of library to use. It was difficult to access which library would perfectly suit my requirements as I was still unsure of the particular steps involved. After research on two or three potential candidates I eventually settled upon the Leaflet library. There were numerous tutorial online that showed how create custom overlays for real Google maps.

The next step was to continue my research by exploring if I would be able to use my own image instead of a Google map one with added effects. The only way to test this would be to create a live local prototype. I started with a basic prototype and then built out of the functionalities one by one. The idea would be just to prove a proof of concept prototype with a rough skeleton of code. By doing this I could eliminate the uncertainty of this functionality.

The first step was to implement the loading of custom images instead of the usual Google map. After getting this working I tested the function with a number of images seeing if I could allow the user to swap the image displayed live. The ability to reload the map and information overlay would be crucial.

Next on the list was the creation of single rectangle color overlaid box of specific size. This proved simple enough lulling me into a false sense of ease. To extend on this I then tried to overlay multiple images over different colors. I quickly ran into problems at this stage. The current setup I had was incorrect for dealing with this kind if functionality.

I was required to go back to the documentation and read up to try and located the required functionality I was looking for. Eventually I found a good tutorial on the website that outlined how to overlay multiple custom images. (Here is the link for those interested http://leafletjs.com/examples/choropleth.html).

The final piece of the puzzle was to understand how to create a popup box that would be custom to each room. This was addressed in an another tutorial. All that was left was to combine each of these elements together to get a very rough prototype.

Doing this exploration allowed me to have a factual base by which to estimate the difficult and assess risk of a previously unknown functionality.

7.3.3 Conclusion

The use of spike can allow members of a team to reduce the uncertainty with implementing features with unknown tools. Technical spike can also be used to carry out feasibility study into any particular suggestion.

This is just one of the many tools that teams should use to try and gain a better understanding of potential product commitments. Often it can be near impossible to estimate to a customer the cost associated with implementing new ideas. Investigating this areas also allows the team to quantify the risk associated with the section. Occasionally ideas might be identified as having a much higher costs associated with them that originally expected. This information can then be given to the customer to ensure they are happy investing their time and money into the specific area.

7.3.4 Sources

- http://agileatlas.org/articles/item/spikes-in-scrum-the-exception-not-the-rule

7.4 Personal blogs

This section of the report contains my weekly personal blogs from throughout the three month process.

Summer research blog – Conor O'Kelly

Sprint 4 – Week 10/11 – 15/8/16

08/26/2016 | OKELLYCONOR | LEAVE A COMMENT |
Sprint 4 – Week 10/11 – 15/8/16

Intro

This will be my final blog post as I detail the completion of coding aspect of the project, presentation and move into the final stages of the completing the report.

Main

During the final two weeks of the we worked to bring the project to a conclusion. This meant finishing of a lot of the features that we had 90% developed in the previous sprint.

My personal work mainly consisted of finishing off the main page for the project. At the end of the pervious week a had gotten very close. Over first three days of the week I worked to finish this off. During this time a make a number of corrections to the main page fixing bugs and other mistakes.

An important feature that I added was making the search bar on the page functional. This meant that users could select the date and building required and the correct information would load on the page. I also implemented variable checking before the user submitted the search request. This meant the user would first need to make valid choices before the search function would execute.

Once this area was finished up it was time to turn my attention towards working on the report. This took up the remaining week and a half of the project.

On the last Wednesday the group had to make their final presentation. We decided to each present the section we felt most responsible for during the development process. We would then each look the innovation we had created. My role at this stage would be to give the audience an overview of the functioning website.

Additional to this we also had to do a 4 minute elevator pitch. I was chosen from my team to present this. We create the presentation as a group on the day and then I deliverd the speech. Overall both of the presentation went well that day.

Conclusion

This is the last personal blog post I would do for the project. The creation of any large scale software is the result of a collaborative effort. Our team worked together consistently throughout the development cycles of this project. Each sprint brought us a step closer to the end goal. As a final note I would like to thank my team for working with me to finish this project together. It was an

enjoyable and educational experience. The software developed here lays good groundwork for future development to roll out to a much large scale implementation. It's a project we can all be truly proud to have taken part in.

Sprint 3 – Week 9 – 8/8/16

08/15/2016 | OKELLYCONOR | LEAVE A COMMENT |
Sprint 3 – Week 9 – 8/8/16

Intro

In this week the group was getting ready for the final prototype presentation and to attempt to wrap up all the coding work for the project this week. The goal as a team was to make an effective presentation on Wednesday where we could show case the majority of our software working. The remainder of the week would then be used to wrap up the project by putting the finishing touches on and try to eliminate as many bugs as possible.

At that start of the week the group still had a significant amount of work to do. We discovered that we had a number of problems running the API and getting everything in sync. One such problem was that a system on the server was incorrectly limiting access to a number of pages.

My work

My main work of the week consisted of the following

- Implementing the Leaflet.js map system for the main page
- Integrating graphs into the main page

The main task for me during this week was the implementation of the leaflet.js system. Overall this turned into quite a lot of testing and javascript. Overall it once I figured out how to implement a number of different features a lot of the work was quite tedious for the overlay.

One of the major problems I encountered when trying to implement the leaflet framework on the page was the interaction between other Javascript on the page. This included issues with different segments load at different times and integrating the framework with the required API calls.

Overall it would have been a much smoother process if was as a group had defined a more strict method for how we where going to code in Javascript.

The second part of my task was to integrate charts created by group member into the main page. This required me to take their code and encapsulate it into new functions that could be called to load the graphs where required. The proved not to be to complicated but a bit time consuming.

Conclusion

During this week we worked hard as a group to push to finish the coding of the project. While we did not achieve this we came very close and are confident that we should be able to finish early next week by Monday or Tuesday. This will five us ample time to create our final presentation and begin the process of writing up the final report.

Sprint 3 – Week 8 – 1/8/16

08/07/2016 | OKELLYCONOR | LEAVE A COMMENT |

Intro

The main goal of this sprint was to try and finished all of the major coding work for the project. Originally the team had hoped to finish all of the coding work by the 14th of August. This would allow us a two week period in which we could continue to do bug test, wrap up loose ends and complete the project report.

As a group and since it was my role as team coordinator I was very much aware that the team still had a log g ground to cover. Our goals of completing all coding work by coding and testing by the 14th still seemed achievable but would set a difficult pace to reach the finish.

There were still a number of difficulties to be overcome including libraries to use to implement our proposed map for the main page and integrating the Android app into our server.

My work

My main work of the week was as follows.

- Creating derived data within the database
- Finished off some the elements of the website and make it look nice
- Starting the implementing of the maps with color overlays on the main page
- Looking into optimizations of the database.

The first of these was the easiest task to complete. It required that a write a script that would derive data for the module and the room table. The script required that each of the whole tables was first taken from the database. I then tried to replace as many of the Null values as possible. For example on the Module table the module level could be derived from the module code. Similarly the floor number that a room was on could be got from the room code no.

The second part of my task which was trying to finish of some of the more minor page on the website took quite a bit more effort. In our haste to start working on the website we had not decided to use a framework like bootstrap. This meant that normally the page would like fine in the size the page was designed in but that it was not very flexible. Instead any changes to the size of the page or the length would results in problems. In hindsight a lot of these issues could have been avoided if we had instead committed to using a specific framework. As it was we decided to just try and fix each issues locally to each page.

As part of our requirement specification we had decided that there would a number of different levels of users to use the product. This meant each of this users would see a different selection of pages. They should also have a unique navigation bar reflection these options.

The final part of my work for this sprint turned out to be by far the most difficult. Our goal was to implement the map show below. This would then have a semi transparent layer overlaid on it. The color of this overlay would indicate to the user how full or empty the room was. When clicked on this would show more information about the room.

(image)

It turned out that this was much more difficult to implement then I had previously expected. There seemed to be only a few options of there. Most of the rest of the week for me was spent researching the different libraries I could used to achieve the results I was looking for. In the end after a small bit of testing I settled on the leaflet.js library. While I did get a little bit started it turned out most of the work for this section was going to spill over into the next week. In the end we had to extend the 3rd sprint by a week to try and finish all of the tasks we had set.

Conclusion

The issues with implementing the map overlay on the main page represents to me a common problem that occurs in creating new software. At that start of the project we had planned to create this information overlays without any investigation into how difficult such a task might be. This did not allow us to properly estimate the amount of time it would take to complete the task. In future it would be much better to perform spike investigations in order to flesh out the feasibility / difficulty of ideas first.

Sprint 3 – Week 7 – 25/7/16

07/31/2016 | OKELLYCONOR | LEAVE A COMMENT |

Intro

During this week it was to be the end of our second sprint and the start of the third. For the third sprint our goals as a group where to finished that data analytics, finished the website will all pages and functionality, have the android application finished, Api implemented and PDF downloadable reports finished.

Obviously this would be a tall order to do all of this work during the two week period.

Personally my task for this sprint would be as follows

- Implementing a feedback system for the data phaser
- Updating the database to meet the new requirements
- Creation of an upload page for the server
- Integrating uploading of files by having he server to call the Python script to upload and then the R script to perform data analytics.
- Implement the map with color overlays for the main page
- Integrate charts into the main page

My work

During this particular week I did not achieving a large amount of my required work. Instead my main focus was on updating the database to fit the new requirements for the data analytics and creation of the upload page.

The first task of updating the database was easy enough. We decided that for the output of our data analytic we would store the information on each class within a separate table. This would be linked by foreign key to a single timetable slot. When looking into the idea I also decided that we should have the table setup in such a way to allow for the estimations to be updated. As part of our innovation we planned to use the Android application to insert new ground truth data into the table.

This over time would allow us to fine tune our model as new information came in. The solution then was to have a timestamp associated with each prediction from the model. By doing this we could then always use the latest estimate for the user while still allow the admin access to historical data.

The second part of my week consisted in building an upload page for the Java server. This would accept files and then store them correctly on the server. This turned out to be slightly more difficult then I had originally anticipated. Since I have not been heavily involved in the deployment and coding of the server side work up until this point. I found the tutorials slightly difficult to follow eventually through the use of stack overflow and numerous hours looking for small mistakes I was able to get the upload page working. As I had experienced previously a lot of the issue that I ran into where a results of a lack of familiarity with using the server. This was another good experience in the importance of staying abreast of the technology that the rest of the team is currently using

Conclusion

During this week I found myself revisiting the issue that occur naturally as each team developed experts. One of the team members was having serious issues with getting the data analytics to work. While she has the models running and producing results our decision to choose R meant that we where finding it difficult to help.

One of the highlights form the week that the benefits of the effort that the team had put into the creation of a solid database schema. The decision at the start of the process to ensure that the database would be in 1st normal form and try to meet all our requirements stood to us. Often any changes required where minor in nature and did not take a lot of time.

This to me represent the benefits of planning elements of any project. While it is important to always have a good overall plan some topics require an extra effort in order to get the details right.

Importance of looking to the future use of a product and trying to plan for them.

Sprint 2 – Week 6 – 18/7/16

07/24/2016 | OKELLYCONOR | LEAVE A COMMENT |
Sprint 2 – Week 6 – 18/7/16

Intro

The main focus for the group was to prepare for the data analytics presentation that was on the Wednesday. As such much of the work was focus on trying to data all of our data in order and ready to present some results. Silvia who had the role of data analytics had already performed her exploratory analysis and was now trying to fine tune the linear regression model with a hope of possibly using a logistic regression model

We had decided as a group in the previous week to use R for our data analytics. I now had to go and learn the basic of R in order to help assist with any issue that might arise. This only took up a small part of my week instead my main focus was on getting the database to auto generate and continuing to update some of the templates on the website.

My week

My main goal of the week was to get the database to auto generate when there is no database. This should then be followed by all the files we got as part of the project being correctly phased with error feedback and input into the database. If files are correctly input they should then be stored one folder. I files fail to be input into the database they should then be moved to a separate folder. This would allows the database admin to review files that might be causing problems and correct the code to deal with these issues in future.

My main task was two fold. The first would be to take the feedback generated from inputing a single file into the DB. Depending on the success of the input move the file. This I found was easy enough to implement.

The second task was to automatically generate the database from file if it did not already exist. This turned out to be the more difficult task for the week but not for the reason I expected. The implementation was simple enough. First the system would check if the database existed. If it did not then the script would use a create the DB by running the SQL commands in the correct file. Each time I ran the script from a blank state the database would be successfully created and then the process would crash. The error message would report that one of the tables (db.Time_table_table) could not be found. This would occur as the process attempting to insert values into the table.

Interestingly when running the process a second time it would first identify that the database had already been created. Skipping the creation process it would be able to succesfully input the raw data into the DB. This time finding the table that had been missing. Between each run no changes would have been made to the database. Instead it would appear the system would be able to find the table after a short break.

Even after much searching I found it difficult to find a solution to my problem. It seemed that there was some internal process that was occurring in MySQL that meant the DB would not be properly mapped if the process attempted to insert information in straight after creating the database.

In the end my solution was to make a SQL request immediately after creating the DB to list all of the tables. This seemed to force MySQL to correctly identify all the tables and it was then possible to insert the data.

Conclusion

Other issues that had occurred during the week was a lack of consistency across the team in the was we interacted with GitHub. This was in terms of the choice of commit and the message we used when committing new code to the database. In retrospect it would have been of great benefit to have done a small online course. This would have allowed each of the member to be on the same page and have a good understanding of what is required of them!

The issue with MySQL gave good insight into one of the bit issues when it come to using external software and libraries. Often when we use this tool we know what we are putting in and what we expect to get out. This high level of abstraction leave us as programmers in a position where it is very difficult to understand what our system is really doing. In order to make full use of our software it is important that we understand the component parts instead of just using them blindly.

Other items also achieved by the group where the creation of a login process for the server!

Sprint 2 – Week 5 – 11/7/16

07/17/2016 | OKELLYCONOR | LEAVE A COMMENT |
Sprint 2 – Week 5 – 11/7/16

Meta

The previous sprint I have summed up in a single post the covers two weeks. I have decided to go back to a weekly approach to posting. I will also try to make my blog post more personal. In previous post I have focused on the overall progress of the team.

Intro to Sprint 2

After the partial success of sprint 1 the team was left in a position where a lot of the initial task of this sprint would require us to revisit work from the previous sprint in order to wrap it up. Additionally we still needed to carry our peer review as a group. This would entail explaining how our code worked to other members of the group.

The main goal aside from this in Sprint 2 was be prepared for the data analytics customer meeting and presentation. This would require that the team begin the process of building our data analytics model that would give us the core information that would be presented to the user.

Secondary to this was to begin the process of fleshing out the website in terms of page and increasing the functionality of the server. Example of this are to include a login system and the provision of an external API that would allow users to gain access (with permission) to our database. This would allow them to carry out their own analytics if the wished.

The provisions of an external API represented one of our innovations. Over the previous two sprint we had given a small bit of though to what innovation we would use in our project. These would aim to go beyond the core of the project and provided some aditional functionality to the user.

Our innovations where to be as follows:

- API
- Android application
- Auto generated PDF reports
- UX testing
- Ability to create custom maps

Main goals

During this week my main goals was to finish off the work I had started in the previous sprint of getting the data input script to correctly work with the database and ensuring all of the information was input correctly. On handy method I had found during this process for putting data into the database was to make sure the db was only updated if the information did not already exist. I used this to try and prevent duplicate occurs.

It was during this process that I had one very frustrating bug that I had to find. When I was running the script some of the timetable information was being input more then once. I ended up spending about 2/3 hours on the Tuesday trying to track this down. In the end it turned out that in a series of

nested loop a single command was incorrectly indented.

While this may seem like a common enough thing to happen I thought it was an important lesson. A lot of the code that I had to sift through in order to find the bug could have done with a decent amount of refactoring. There where too many sections of code that where too long. Trying to keep a mental recored of what was happening and trying to track down errors when the information was incorrect became painful. Overall this was an important lesson in ensure that I try and break down my code in smaller and more clear sections. This makes it easier to validate the output.

The team

It was during this week and the end of the previous week that a number of group issues came to my attention. Being the group coordinator to me meant that is my job to ensure that the group worked together as a cohesive unit. At the start of the week we discussed any issue the group had experienced during the pervious sprint. One of the main issue turned out to be with the devision of work among the members

Half of the team felt the other half had taken on too much coding work alone and that they had ended up being stuck doing some of the less interesting and challenging design work. I tried to address the issue as best as possible and made a commitment that we as a group would try to prevent such occurring in the future.

Week end

The end of the week we set out plans as a group to try and begin the process of fleshing out the website. This would mean building the pages that where required and mapping them on the server. This would allows us to have rough template so we could being implementing the features.

One of the main issue that I personally experienced was my lack of knowledge with using the Java server. Having not been the one to set it up it required me to go back and review all of the documentation.

These types of issues normally seem to occur in group project like this. It seems there is a tendency for experts to develop in different field. This occurs as each member has more focus on their section of the project and develops their skillet. In the end in order to catch up on using the Spring server I followed the documentation online to build my own basic Hello World server. This hands on experience gave me enough information to move forward and try and understand the other elements of the server that I have not been part of implementing.

Week 3/4 – Sprint 1 => Initial architecture setup

07/11/2016 | OKELLYCONOR | LEAVE A COMMENT |
Week 3/4 – Sprint 1 (27/6/16 – 10/7/16)

After setting out the main requirements last week it was now time for the group to make a number of decision on how the features were to be implemented. The main ones would be the choice of database and the database design, the choice of server environment and library used to run the server, process to clean the data. and more strict coding design constraints.

In this sprint our plan as a group was to make educated decision on all of the technologies we use for the project. These would take into account some of the later requirements of the project. A key example for this was out plan to build an android application. When comparing Spring to Flask we find Spring already has support built in for Android app development.

Architecture choices – Spring server environment

One of the corner stones of the project would be the choice of the server environment we would use. In the initial introduction week where where show how to set up a basic Tomcat server using Java. We felt that we should first explore all of the options before commiting to server.

As most of the class we found two accessible choice when it came to language. This was Python or Java. Most of the class was already familiar with both but more so with Python as many of the group has used it in their Dublin bikes project along with Flask.

We as a group weigh up the two main options for implementing our server. These where using Flask in Python or using Spring in Java. The Flask server would be initially easier to setup and get working. In comparison we as a group had no experience using a Spring server. The benefits of going down the Spring route would mean creating a product that would scale better and have significantly more features. We also felt the a Spring server would allow the group to gain a broader range of skills and look more professional.

Early on as a group we had decided that on of our innovations would be the creation of an Android application. Spring comes ready to integrate with Android application and this in the end heavily influenced our decision.

In the end we decided to go with a Spring server. This could handle the theoretically large demand required to role out our product while providing experience in well regarded library.

Architecture choices – MySQL database and database design

The choice of database for any project is an important one. It will act as one of the main focal point for the system. The choice and design of a good database would set a good foundation. By investing time and effort at this point of the project it would save us having to revisit the database schema to make updates throughout the project.

As a group we wanted to choose a database that would meet all of our requirements and give us experience with working with professional tools. When we began our search there where a number of options to choose from. The main ones where MongoDB, sqlLite and MySQL.

In our research we discovered the one of primary measures of a Database is the its transactions have ACID properties. These are Atomicity, Consistency, Isolation and Durability. The are fundamentals required for a reliable a transaction system.

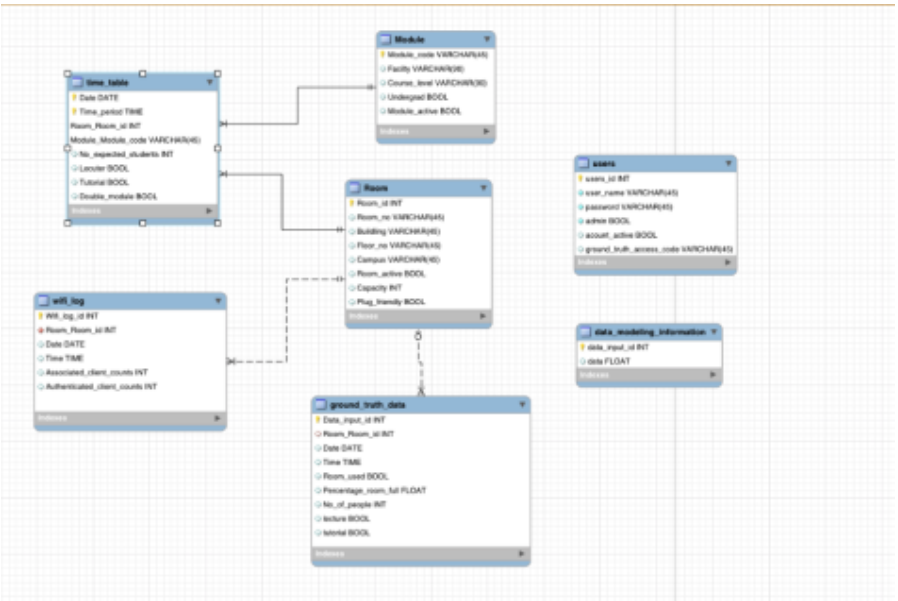
Our project didn't consist of any super important information and the loss of a few transactions might be considered acceptable. Despite this we felt is was important to explore the options between databases to try and understand their benefits and limitations

The most interesting database as MongoDB. This is widely regarded in the industry as being much faster then a traditional SQL database. The down side of this is that it is not fully acid complaint and this can result in a number of strange bugs or quirks of the database. It was due to this lack of compliance that we ended up ruling out MongoDB.

Instead our ultimate decision came down between SQLite and MySQL. The main difference between the two was that SQLite was server-less and MySQL required an internal server to be setup. In the end we opted for MySQL. The main reason behind this decision where that we felt it would be more professional and MySQL would allow better scaling for a larger project.

This then lead us to the process of designing the database schema. This was a part of the project that I took particular interest in. I initially worked alone to produce a rough schema that tried to encompass all of the requirements of the project. This was then presented to the group to review. Going over each section we managed to flesh out the tables adding in columns and connecting them appropriately with foreign key relationships.

I then took the database and converted it into first normal form. This was the end results



Data cleaning

We decided that the easiest method of cleaning the data would be to use a Python. Python is well recognized for its flexibility when it comes to this area. The majority of the difficult to process data came in the form of Excel files. This where often badly formatted and had a number of error in the way that they had been laid out.

One of the more time consuming issues that I encountered at this stage was dealing with already merged cells excel. A number of libraries would treat a merged cell two cells of consisting of one cell with all the information and the second cell as being empty. Often the merged cells would identify double module. When processing the data initially I found that the information from the second half of a double module was being lost. This required a decent bit of work to design a function that would un-merge all the cells in the document while preserving the information.

During this process my goal was to attempt to design the data cleaning section in as modular a format as possible. To do this I would create a different module for each of the different data types that I would be receiving. This would then all be tied together will a single data input manager. This would received the processed data from each of the files and insert it into the database. The design practice would allow for flexibility when changing how the data was processed.

Coding guidelines

When working to together as a team we felt that it would be important that everyone was on the same page. In term of code this would mean that everyone would code in a similar style. This make it easy for one person to transfer part of their work to another person and for peer review to take place.

For Python we decided to use PEP8 as out style guide. This had many specific rules on how we should format our code and how the overall script should look. Conveniently automatic checkers could be added to many of the IDE's that the group planned on using.

For Java we decided to follow the code guidelines set out by Google.

Working towards our first prototype

It was during this period of time I personally would have to present on the first prototype that group had. This would show case the work that we had done.

The main goals for this presentation where as follows.

- Database implemented
- Date input manager running
- Wireframe for website creating / Website style guide
- Site map for website created
- Server up and running with initial hello page

Conclusion

Overall this sprint provided some valuable lessons in working together as a team. A significant issue that came up was the devision of work among the team. Half of the team felt that has spent to much time doing designing for this sprint while the other half of the team spent the time coding. This would require consideration going forward that we did not have repeat situation.

Secondly to this was the progress of the overall sprint and the completion of work. We found that when we went back and reviewed our goals for the sprint we had by and large achieved them. Most of the functionality we had aimed for had been implemented. Unfortunately the rush to getting most of this finished meant a lot of the code was still very rough and did not have adequate testing. This would require us to revisit a lot of this work in sprint 2 in order to bring it up to standard.

This represented a second important lesson in ensure that not only was work finished but that it was finished correctly and to standard.

Sprint 0 – Week 2 => Planning

06/26/2016 | OKELLYCONOR | LEAVE A COMMENT |
Week 2 – Sprint 0 (20/6/16 – 26/6/16)

This was the second main week of our project. The main focus after this week was to choose the methodologies that we would try to implement throughout the project. As we had covered in class and as most group were probably going to go our aim as to roughly implement a form of Agile development known as scrum. With most of us having used this development framework in earlier

project we felt it should be easy enough. As my role as coordinator it would be my job to do high level management of the process. The first step to this would be to being the first scrum. Being CS students we decided to name this sprint 0.

Our goal in the first sprint was to flesh out the requirement of the project. The brief for requirements on the project while specific left a large amount of room for scope. The secondly goal of this sprint and possibly equally important was to agree upon what technologies we as a group would be using and how we would use them. This sprint was planned to (and did) last only from the 27/6 – 3/7. The aim was to set the ground work from which most of the project would grow.

Team work tech choices

In term of tech choices we decided as a group to use the following;

- Google drive to host all of the group work
- Github as a version control
- Google docs to host all our collaborative documentation.
- Appear.in to meet (instead of using skype)

Along with this I would also run a minutes for every meeting. The goal would be to set an agenda before / at the start of a meeting, take minutes as the meeting progressed and then write a conclusion to summaries the main points. This would act to control meetings and crate a focus point where anyone in the group could check back to refresh recent discussion and review what the current goals of the group would be.

Main goals of the week and actualization

The main goal of the week was to flesh out the requirement of the project in terms of the SRS. The idea behind this was so that all the members of the group would have a base framework for what we where trying to achieve. The process we used for this was to divided each of the sections of the SRS up and fill them out. Then we came back reviewed each other sections and tried to add or improve the area.

The secondary goal of the week was to establish the framework in which we work in. This meant starting the process of exploring the technologies we would use to bring the project to life. We realized that getting the requirements right at this stage was very important.

These areas included the following:

- Database
  - Sql vs sqlLite
  - System for interacting with the database
- Server framework and server language
  - Java vs Python
  - Java MVC?
- Language for data cleaning
- Data analytics tools
- Software for organization (like Maven)
- Formatting of code
- Coding standards (like Pep8)

As a team we would try and focus on creating clear and well documented code. This would aim to be a modular as possible. We would have to wait and see if our ideals would survive the reality of deadlines!

During this process we also looked the what our future innovations might be and how we might realize them.

Requirements

The following where to act as our core requirements.

- Data analysis
  - Calculation of no of people in the room
  - Recalculation of averages over time
- Users interfaces
  - Website
    - Use of a map to show the location of rooms within the building
    - Used to show averages as well
    - Present information all of the data analysis available
      - Possibility of two format => simple / full information
    - Slider allowing user to see how occupancy changes over time on a particular map
      - Colour coded / percentages ?
      - Good high level overview
  - Mobile app
    - Allow users to select the room / time
      - In order to input ground truth data
    - This will then be confirmed as being input
    - Filter for nonsense information at server level.
- Technical requirements => brief overview of the architecture
  - Server
    - Manage interactions between interfaces
    - Ensure security
  - Database
    - Use of full sql database to manage all the information
- Management requirements
  - Ability to change what rooms are timetables
    - System should be able to interface with the data format provided by ucd
  - Ability to change what rooms are being monitored (add / remove)
    - Included adding / removing maps
  - Change meta information about rooms
  - Upload wifi data in a specified format
  - Ability manually add ground truth data through the website
    - Can be done a specified format
    - Can be manually added for each room
- Data output
  - Production of reports
    - PDF reports that are auto generated with graphs
    - Could be specific to room / module
  - Users can download bulk information
- Documentation requirements
  - Specify methods for carrying out interactions with the database
  - Template for inputting information of all kinds

- Data cleaning for input requirements
  - Each area where information is taken in form the user / management
  - Data will be cleaned and formatted as specified by the documentation
  - This will involve small bridging scripts that will be unique to each location
  - Alternatively we will also allow users to input the data directly.
    - Will also allow for updates

- Extra features
  - Ability for users to change the maps format
  - Allow lectures to contact other members of staff requesting a room swap

Future sprint planning

At the end of the week as sprint 0 came to a close and we completed the requirements documents on our vision of what the project might entail we focused on creating a roadmap. This would act to place all of the features for the project within a series of sprints.

Our main goal was to have the code elements of the project finished by the 14th of August allowing us ample time for delays and give us two week to wrap up the write up and presentation of the project.

Our roadmap was as follows.

- Sprint 0 => Week 2
- Sprint 1 => 3-4 (Rough working prototype – Server / Database / basic website / cleaning scripts)
- Sprint 2 => 5-6 (Data management / Data modeling / model selection / server APIs / Website presentation)
- Sprint 3 => 7-8 (Extra features of the website / UX of website / Android app development)
- Sprint 4 => 9 (Deadline for project to be finished)
- Sprint 5 => 10 (Debugging / Extra issues / overfill week)

Conclusion

During this week we laid many of the foundations for the project. These would act to guide the project to it own conclusion while allowing us the flexibility to respond to change. Take an agile development methodology we tried not to focus to much on the creation of point less documentation. Instead we aimed to give a face to the core requirements of the project and to flesh out how these might be achieved.

From this blog post on I plan not to write as long posts but instead to provided summaries as we progress through the sprints.

My approach to personal blogs

06/19/2016 | OKELLYCONOR | LEAVE A COMMENT |

My overall approach to writing my personal blog my not be a conventional one. I have decided that since I am the team coordinator that role should influence the direction and content of the blog. Since my main role is to facilitate the overall direction of the team this will be one of my main topics! I will also review at the end over every week how I felt both myself and the group is getting on with progress.

Week 1 Summary

06/19/2016 | OKELLYCONOR | LEAVE A COMMENT |

During the first week of the project we as a group got off a relaxed start. The aim of the first we was to try and build relationships as a group and to get used to the technology that we planned on using. The Wednesday goal of this week would be for one member of our group to do a presentation on the big ideas of our project as it currently stood.

During the first meeting that we has scheduled to have the Monday the main focus to sit down and brain storm different ideas on the project. We had each left the previous Friday with the intention of going home and thinking about the potential vision we has for this software. Since input from the customer was quite scare at this point (and potentially for the rest of the project) it would be up to use to try and create a unique project the offered high utility to the user. Topics that where discussed included server language, database design, general features of the software and extra features that would separate out project out from the rest.

Each of the group members gave good input in this meeting and we managed to make good progress. One of the more interesting ideas that came up was to include the creation of an android application for the input of ground truth data. We later learned on the Wednesday from the customer that this is something they where very keen about. The application would allow the user to select the room and time and then input the number of students that where currently there. This would allow lectures to easily add this data. Information on the ground would also allow us to correct our model over time.

For the data analytics sections of this software we fleshed out the ideas of two different approaches. The first would be to get ratio estimates for number of devices to number of users based on the historical data. The second approach would be to use the ground truth data to correct the model over time. This would start at a ratio based on the historical data. As new information came in the model would correct itself.

Another interesting topic of discussion that also came up was why type of database to use. I personally had a number of strong opinions on the issue after doing a small bit of research. Initially it was suggested to avoid the use of a full SQL DB and instead look at some of the more light weight databases. I was against the use of an SQL lite database mainly due to already experiencing search time issue on a pervious project when I had used a large data set for one particular table. Their where also suggestions of using MongoDB. While in this might had fulfilled that basic requirements of our project I was also against using an immature technology that I felt Mongo DB represented. I will have further blog post on this particular issue where I try and explore it in more detail.

On a more personal note this was to be my first week to trying to maintain the relevant coordination documentation. I felt that as my job as team lead I would try and ensure that all the meetings would have minutes taken for them. I had hoped this would provided a clear path of how the meetings and overall goals of the project progressed. As I look back over my notes for the first week I can see a clear progression to more organised layout for these documents. With an aim to create informative titles for each included the date as a start. I eventually settled on a layout of an agenda, minutes and conclusion. The agenda provided an area for the group members to state topics they felt needed discussion during out meeting. It would also help to provide focus for out meetings. The minutes

would provide a running commentary as the discussion progressed. Finally the conclusion would allow all the group members to come back and easily remind themselves what had been discussed and agreed to.

During each of these meeting I felt that is was my job as group coordinator to try and keep these meetings on track and facilitate the discussion between members of the group. I have also created a group diary where I have asked each member of the group to give rough outline of the work that they have completed on a daily basis. This can allow us as a group to provide a timeline of our progress.

After my initial apprehension with being coordinator for the group I have started to settle more into the role. Using the agile methodology I have come to realise I must move away from a focus of excessive documentation. Instead it is important to develop focus for the group on a common vision that we can work cohesively together towards.

The next week will be week 2 (20/6/16) of our project. During this week we hope to being to iron out all of the features of our software and to write up a basic version of the SRS. This will act to set the foundation of out project. It will also be the zero sprint and will consist of mainly planning. After this week I hope we will get back to more coding and getting stuck into building the actual software!

In future I hope of have more regular and small blog post. These I feel will better allow me to explore some of the more specific issue while they are still fresh in my memory.

Introduction Post

06/14/2016 | OKELLYCONOR | LEAVE A COMMENT |
In this blog will cover the progress I make on my summer research project as part of my UCD Computer Science conversion MSc. The aim of the project is to work together as a team to create software that attempts to accurately predict occupancy of a room by the number of devices connected to the WIFI.

Throughout the summer I will be working with my team as we design, build and test your ideas on the path to creating a good software.

During this process we will be implementing Agile methodology in order to manage the multiple different aspect of the project. We use a number of different tools in order to keep track of progress and also to measure overall quality.

Each team member will have individual roles throughout the project. Each roles will require the team member to take responsibility for a different aspect of the project. I have originally hoped to get the role of code lead. I felt that I would be strongest in this area. Instead I ended up receiving my second choice of coordination lead.

While initially I would have been disappointed with this choice I believe this role will allow me to develop a broader skill set in the discipline of software engineering. I will still being able to strongly take part in the coding aspect of the team I will also be able to learn about the issues of team management and communication. Working individually is normally easy enough. My goals will be to ensure we work effectively together while minimise the bureaucratic overhead.

References

[1] R. Melfi, B. Rosenblum, B. Nordman, and K. Christensen, "Measuring building occupancy using existing network infrastructure," *2011 International Green Computing Conference and Workshops*, 2011.

[2] A. Sevtsuk, S. Huang, F. Calabrese, and C. Ratti, "Mapping the mit campus in real time using wifi," *The Practice and Promise of the Real-Time City*, pp. 326–338.

[3] S. Zikos, A. Tsolakis, D. Meskos, A. Tryferidis, and D. Tzovaras, "Conditional random fields - based approach for real-time building occupancy estimation with multi-sensory networks," *Automation in Construction*, vol. 68, pp. 128–145, 2016.

[4] A. M. Saandeep Depatla and Y. Mostofi, "Occupancy estimation using only wifi power measurements," *IEEE Journal on Selected Areas in Communications*, 2015.

[5] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth," *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014.

[6] T. Matsunaka, A. Yamada, and A. Kubota, "Passive os fingerprinting by dns traffic analysis," *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, 2013.

[7] A. F. Zuur, E. N. Ieno, and A. A. Saveliev, *Mixed Effects Models and Extensions in Ecology with R*. Springer, 2009.

[8] "R leads rapidminer, python catches up, big data tools grow, spark ignites." `http://www.kdnuggets.com/2015/05/poll-r-rapidminer-python-big-data-spark.html`. Accessed: 2016-08-25.

[9] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 6. Springer, 2013.

[10] spring.io, "Resttemplate module." `http://docs.spring.io/autorepo/docs/spring-android/1.0.x/reference/html/rest-template.html` accessed 21 August 2016.